NOTE TO USERS

This reproduction is the best copy available.

UMI®

University of Alberta

A Bayesian Theory for Forestry Image Understanding

by

Li Cheng

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta
Fall 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-95917-1
Our file Notre référence
ISBN: 0-612-95917-1

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To mom and dad
for their support

Preface

Perhaps the best way of documenting how expert photo interpreters interpret forestry images is to examine what they are taught when being trained. As summarized in Hall [42], the photo interpretation (PI) procedure has the following components:

- prior knowledge of terrain and of forestry (stand and individual tree characteristics) is critical
- Depth information about trees
- Understanding of image clues is essential. Important image clues include: shadow, tone (spectra/contrast), texture (types of canopies), shape (canopy shape), size (canopy size), species distribution and composition — all related to the knowledge of the terrain and of the forestry
- Reasoning about what is perceived with what is known: integrating 2D image clues with 3D models and vice-versa
- Progresses from what is clearly interpretable to the less clear.

In Chapter 1 we survey the existing forestry inventory approaches, and the analysis of the drawbacks leading to the development of a novel Forestry Image Understanding System (FORIUS). A review of the related statistical background is given in Chapter 2. We propose our FORIUS framework in Chapter 3 and then describe in details the components in Chapters 4, 5 and 6. Experimental results are shown and discussed in Chapter 7. Finally, conclusions are drawn and suggestions for future work are given in Chapter 8.

Contents

1 A Review of Existing Forestry Inventory Systems 1
- 1.1 Image Variation Based (Model-free) Approaches . . . 2
- 1.2 Template-based Approaches . . . 4
- 1.3 Summary and the Proposed Framework . . . 6

2 Related Statistical Models and Methods 8
- 2.1 Bayesian Analysis . . . 9
- 2.2 EM algorithm and the variants . . . 13
  - 2.2.1 Introduction . . . 13
  - 2.2.2 The EM Algorithm . . . 13
  - 2.2.3 Variational EM . . . 15
  - 2.2.4 Variational Bayes . . . 19
  - 2.2.5 Upgrading the Bound on EM . . . 22
  - 2.2.6 Stochastic EM . . . 24
- 2.3 The Sampling Algorithms . . . 26
  - 2.3.1 Direct Sampling . . . 26
  - 2.3.2 Importance Sampling and Rejection Sampling . . . 27
  - 2.3.3 Markov Chain Monte Carlo (MCMC) . . . 28
  - 2.3.4 Model Selection with MCMC . . . 31
  - 2.3.5 The Roles of MCMC and Deterministic Algorithms in Approximation . . . 33
- 2.4 HMRF Model . . . 35
  - 2.4.1 the HMRF Model . . . 35
  - 2.4.2 Relationships with the Boltzmann Machine and the Hidden Markov Model . . . 38
  - 2.4.3 Algorithms for Parameter Estimation . . . 39
  - 2.4.4 The Mean Field Algorithms . . . 41
- 2.5 Summary . . . 44

3 The Segmentation and Labelling Component 45
- 3.1 The Image Model: Hierarchical Markovian Models . . . 46
- 3.2 The Hierarchical Hidden Markov Tree Model . . . 48
  - 3.2.1 The Estimation Algorithm . . . 51
  - 3.2.2 The Segmentation and Labelling Algorithm . . . 52
- 3.3 Performance Evaluation . . . 54
- 3.4 Initial Comparison with Similar Methods . . . 55
- 3.5 Forestry Data Analysis . . . 56

3.6 Summary 58

4 The Stereo Component **62**
4.1 The stereo model 64
4.1.1 The Data and Prior Models 65
4.2 The Inference and the Learning Tasks 67
4.2.1 LBP for Approximate Inference of d 67
4.2.2 Updating σ_y and σ_d 68
4.2.3 Updating s_y 68
4.2.4 Updating s_d 69
4.2.5 the Updating Strategy 70
4.3 Experimental Results 70
4.4 Discussion 73

5 The Tree Generation and Fitting **75**
5.1 Problem Formulation and Notation 77
5.2 The Stochastic Expectation Maximization (SEM) Algorithm 80
5.2.1 One 3D Tree per Segment 80
5.2.2 Known Number of 3D Trees per Segment 82
5.2.3 Unknown Number of 3D Trees per Segment 83
5.3 The Jump–Diffusion Sampling Paradigm 84
5.3.1 Problem Formulation 84
5.3.2 the Proposed Algorithm 85
5.3.3 The Update Move 86
5.3.4 The Merge and Split Moves 86
5.4 Implementation and Experiments 87
5.5 Summary 89

6 Experimental Results **90**
6.1 Data One: Evaluation of the Segmenter-Annotator 91
6.2 Data two: Evaluation of the Synthetic Forestry Images at different resolutions 92
6.3 Data Three: Evaluation of the Synthetic Forestry Images with Issue of Under-segmentation 93
6.4 Data Four: Evaluation of the Timberline Image Dataset 93
6.5 Summary 94

7 Conclusion and Future Work **105**

Bibliography **109**

A Appendix for Chapter 3.6 **115**
A.1 Nomenclature of the segmentation and labelling component 116

B Appendix for Chapter 2.5 **117**
B.1 The Derivation of Eq. (2.58) of the Stochastic EM 118
B.2 Proof of Eq. (2.16) 118
B.3 Proof of the Equivalence of Two Lower Bounds 119
B.4 Derivations that Related to Variational Bayes (VB) 119

B.5 Proof of the M–H algorithm 122
B.6 Definition of Markov Random Field (MRF) and the Hammersley-Clifford Theorem 122
B.7 Derivation of the Log-partition Functions 123
B.8 Derivation of the Updating Equations in Exact–EM 124
B.9 Derivation of the Updating Equations in NMF–EM 127

C Appendix for Chapter 4.4 **129**
C.1 Full Conditional Probability 130
C.2 Updating σ_y 130
C.3 Estimating the Partition Function $z(p_d)$ 131

List of Figures

1.1 This figure shows the basic process-flow of the template-based approaches. After aerial images are taken, templates are generated by rendering the 3D super-ellipsoids into 2D images, and are then used to cross-correlate images to detect optimal matches. 4

1.2 Proposed forestry inventory framework. The three sub-processes are the **segmenter-annotator** module to generate labelled regions, the **stereo** module to generate depth map, and the **3D trees fitting** module to produce plausible scene geometry. 7

2.1 This figure shows the probabilistic graphical model interpretation for applying the Bayesian approach to the missing data scenarios (see text for details). 10

2.2 The diagram schematically shows, as function of the parameters θ, one log-evidence function $L(\theta; y)$ and its lower bound $\mathcal{L}(\theta, q)$. 14

2.3 This shows the variational EM interpreted as the first-order bounding. 18

2.4 An exemplar 3×3 HMRF model. The white nodes are the latent variables x, and the grey nodes are the observed variables y. Associated with any edge e connecting two neighbouring latent nodes (x_i, x_j), there is the function represented as $\phi(x_e)$. Similarly, for any (x_i, y_i) pair there is an associated function represented as $\phi(x_i, y_i)$. 37

2.5 An example of the GMF algorithm. Here we show a 3×4 2D graph of the HMRF. By applying the GMF algorithm, the graph is divided into 6 subgraphs, with each subgraph of the size 2×1, which can be regarded as a meta-node. See text for details. 44

3.1 The basic Hierarchical Hidden Markov Tree (HHMT) model. Here only three levels $(l-1,\ l, l+1)$ of the multiscale representation are shown. At each level there are two random fields corresponding to observed pixels, Y, image region labels defined by X. The bijection operations shown in grey and black colors (with corresponding arrows) represent upward and downward support kernel operators and associated region label transitions over movements upwards and downwards in scale, encapsulating two types of contextual constraints. 48

3.2 A simple example of the indexed histogram codebook, with a window of size 3×3. Assume we have two class of labels, e.g. $C = \{1, 2\}$, at all levels, and we are in layer l. The numbers in the two most left columns in (a) and (b) represent the frequencies of class 1 and 2 for level $l+1$ (a) over the image while the probabilities correspond to the relative (conditional) frequencies of occurrence of labels 1 and 2 at the next level l (a). (a) is the table for downward operation, and (b) is the table for the upward operation. 54

3.3 A comparison of performance of the TSMAP algorithm and the proposed HHMT method on grey-scale document images. (a) and (b) show example training images. (c) shows a cropped part of one of the test images. (d) shows performance of the TSMAP model while (e-f) show performance of the HHMT method (see text for more details). 56

3.4 (a) Shows a simple regular spruce plot with some mixtures with aspen (left) on a sunny day. (b) Shows a complex plot consisting of a mixture of aspen and spruce on a sunny day. (c) Shows a simple regular spruce plot in cloudy day, and (d) shows an example of an early Fall image. 57

3.5 Shows performance on three types of forestry conditions: (a) Spruce only; (b) medium mixture of spruce and aspen; (c) significantly overlapping mixtures. Second row: classification by Naive Bayesian classifier. Third row: segmentation and labelling via the HHMT method. Forth row: expert annotation. 59

3.6 The HHMT model was estimated using image (a) and tested on images (a) and (d). We have used 4 classes corresponding to Aspen, Spruce, Shadows (both the shadow on the trees and on the ground) and Ground. Labelling results are shown using different colours. (b) and (e) show labelling after the first iteration (i.e., Naive Bayes Classifier) while (c) and (f) after five iterations. The bottom graph (g) shows the similarity scores (average log MAP probabilities per image) for the image data given the model as a function of the number of iterations of the algorithm. 60

3.7 Shows more detailed examples of how the HHMT model fits expert annotation of spruce regions for the images in Figure 3.5. 61

4.1 The proposed stereo model. The grey circle denotes the observable variable, y, corresponding to the disparity space computed from the left and right images and is the same size of the Markov Random Field (MRF), d, with each site containing a vector of size D. The white circles correspond to unobservable variables. The node d represents the disparity map. The σs refer to scale parameters and the ss refer to shape parameters. 64

4.2 Examples of the potential function of the Generalized Gaussian distribution. Left: non-convex function when taking the shape value $s = 0.7$; Right: convex Gaussian function with the shape value $s = 2$ and the scale value $\sigma = 1$ for both cases. . . 66

4.3 Experimental results for the "sawtooth" stereo pair. (a) shows the left view of the sawtooth stereo pair. When only employing the LBP algorithm for the inference of disparity map, (b) shows the resultant disparity map with fixed hyper-parameter values $(\sigma_d, \sigma_y, s_d, s_y) = (2,6,2,2)$, and (c) is the resultant disparity map with another fixed parameter values $(\sigma_d, \sigma_y, s_d, s_y) = (5,10,2,2)$. When the MCMC sampling techniques are employed, despite the different starting values as in (b) and (c), after a few iterations, hyper-parameters end up converging to fixed-point (0.01, 2.80, 0.73, 0.99), with the fixed-point disparity map inferred as shown in (d). 71

4.4 Experiments on the "map" stereo pair. (a) shows the left view of the stereo pair; (b) shows the disparity map with fixed parameter values $(\sigma_d, \sigma_y, s_d, s_y) = (2,6, 2, 2)$; (c) is the inferred disparity map by learning the scale parameters for Gaussian functions; (d) is the inferred disparity map by learning both the scale and the shape parameters for Generalized Gaussian functions. 72

4.5 The first column shows the resultant disparity maps for the four image pairs using the Generalized Gaussian model, and the second column shows the corresponding predicted disparity error compared to the ground truth. Here black pixels are counted as errors while gray pixels are in occluded patches and not counted. . . 73
4.6 The first row shows the evaluation errors for the four image pairs using the Generalized Gaussian model. The second row is the corresponding errors using the simple Gaussian model. Both are acquired with the aid of the online evaluation webpage that can be accessed via http://bj.middlebury.edu/~schar/stereo/web/eval.php. (See text for explanation of the ranking orders in the brackets.) 74
4.7 Experiment on one of the synthetic forest image pairs using the Generalized Gaussian model. (a),(b) show the left and side view of the trees — eight aspens and eight spruces. (c) is the disparity map of (a) at $t = 0$ step, and (d) is the disparity map at $t = 10$ step. 74

5.1 This example shows the typical results of the 3D tree fitting module. Top left is the left view of the input image pairs that contains two tree species — aspen and spruce. The bottom left shows the side view of the input scene geometry and the bottom right show the corresponding preprocessed data. This is the grey-scale depth map where the lighter the pixel value, the closer the object at that pixel to the viewer (camera). The 3D tree fitting module takes the preprocessed data and infers the latent 3D scene geometry, which is rendered back to 2D image with coloured texture mapped for different tree species, as shown in the top right. . . 77
5.2 This figure contains a) the main graphical model of the proposed 3D tree fitting module, and b), on the right, the zoom-in view of the graphical model inside node S. 79
5.3 This shows a simple case of Fig. 5.2 when $K = 1$. 79
5.4 This figure shows the error measurements of the 3D tree volume (red curved surface) versus the corresponding depth map (blue surfaces). 81
5.5 This figure shows a group of 3D super-ellipsoids with each sitting on top of a labelled 2D tree region. Notice in practice, the error measurements of the labelled regions are on 2D image plane with which the 3D trees rendered onto. 81
5.6 A typical simulation run on the testbed data. Top left: the left view of the input image pair. Bottom left: the labelled regions. Bottom right: the inferred scene geometry using initial estimated parameters, with one tree per segment. Top right: the final result after running the proposed JD–MCMC algorithm to convergence. 88
5.7 Given one segment, started from two 3D trees, this shows the dynamics of the fitting process. Top left: the left view of the input image pair. Bottom left: Initially inferred scene geometry. Bottom right: result of the split move at the 5th iteration. Top right: result after 40 iterations. 88

6.1 An example of the experiments on airborne grey-scale image data of around 100 cm per pixel. (a) The input cut-plot image. (b) The segmentation result of (a) with intersection/union confidence 0.5350 (includes the tree and the shadow classes). (c) The input cut-plot image. (d) The segmentation result trained on (a), with intersection/union confidence 0.3191 (includes the tree class and the shadow class). (e) The input cut-plot image. (f) segmentation result trained on (a), with intersection/union confidence 0.3653 (includes the tree class and the shadow class). Here we used the average colour for each segmented regions. The image consists of three classes, while green colour is for tree pixels, black colour is for shadows and brown colour is for the remaining pixels. 95

6.2 An example of the experiment on airborne colour data of about 30 cm per pixel. (a) the training image. (b) The expert labelling result. (c) The testing image. (d) The segmentation result with intersection/union confidence 0.3726 (includes two tree types only) or 0.2103 (includes two trees types and the shadows). The image consists of four classes. In the expert labelling image, the green and yellow colours are labelled to identify the pixels of the two tree types, respectively, blue colour is for shadow pixels and the remaining is left unmarked. 96

6.3 This shows the training image used with 20cm/pixel resolution. In expert labelling of this set of experiments, red colour is marked for aspen tree, green colour is for spruce tree, blue colour is for shadow and the remaining is left unmarked. . . . 96

6.4 Plot of 16 trees (left two columns of spruce, right two of aspen) with equivalent image resolutions of (20,30,40) cms/pixel. 97

6.5 Plot of 16 tree (left two columns of spruce, right two of aspen) with equivalent image resolutions of (60,80,100) cms/pixel. 98

6.6 The simulation results for the resolution of 20cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result. (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. 98

6.7 The simulation results for the resolution of 30cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. . . 99

6.8 The simulation results for the resolution of 40cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. . . 99

6.9 The simulation results for the resolution of 60cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. . . 100

6.10 The simulation results for the resolution of 80cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. . . 100

6.11 The simulation results for the resolution of 100cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry. 101

6.12 This figure shows detection and labelling of trees as a function of image resolution, where the function values equally represent the corresponding confidences. (solid line: Spruce, dashed:Aspen). 101

6.13 This figure shows the robustness of the 3D fitting module when the images are under-segmented. The left view of the input images pair is shown in a), and its segmentation result is shown in b) where several tree regions are merged together due to the overlapping of these trees in the input image. c) and d) show the initial geometric scene when start from $K^{(0)} = 1$ and $K^{(0)} = 2$ per segment, respectively. e) and f) show the corresponding intermediate results at iteration 10. g) and h) show the corresponding final geometric scene results. 102

6.14 This figure shows the corresponding side-view of the 3D fitting dynamic in Fig. 6.13. The left view of the input images pair is shown in a), and its side-view is shown in b). Starting from fitting one tree per segment as shown in c), the side-view of the scene geometry is shown in e); after convergence, the side-view of the scene geometry is shown is g). Similarly, d) shows the initial fitting scene geometry with the side-view shown in f), and h) shows side-view of the final scene geometry, respectively. 103

6.15 This figure shows the experiment of applying the FORIUS system to a real forestry plot from Timberline dataset. a) and b) show the left and right views of the forestry image pair. The white box of a) and b) form the input image pairs, as shown in c) and d), respectively. e) shows the segmentation result of the left view c). f) shows the disparity map of the left view c). The 3D fitting module starts from g) and ends at h). 104

Chapter 1

A Review of Existing Forestry Inventory Systems

As in Gillis and Leckie [35], the task of forest inventory could be summarized as the collection of select information for a given forestry plot, including

- species composition
- location of each tree crown, crown closure (the segmented tree crown region), crown size
- stem height and tree volumes.

This information can be extracted from remotely sensed imagery data. The traditional approach [35] involves expert photo interpreters analyzing the image data to

- detect and delineate individual tree crowns, tree volumes, and
- determine tree species.

These results, plus the amount of prior knowledge for the plot of interest, become crucial for any further analysis.

However, this approach involves enormous human labour, and even visual interpretation has suffered from the criticism of being subjective and not accurate. With the recent advance of sensor technologies, it is becoming affordable to get access to multi-spectral, geo-referenced, high-resolution (typically below one meter) airborne (or even satellite-borne) image data [50]. This enables research and development toward an automated forestry inventory system.

Over the past decade, many research projects have been conducted [15]. The resulting systems can be categorized into two groups: (a) image variation based, and (b) template based approaches. Approaches in (a) tend to directly analyze the local convexity of the image data. On the contrary, approaches in group (b) feature a pool of 2D tree templates that generated from a 3D tree-modelling component, under various lighting and geometry conditions.

1.1 Image Variation Based (Model-free) Approaches

Such approaches, e.g., [68] [37] [16] [70], being mainly developed for grey-scale images, are based on the detection of contrast between the tree crown and the surrounding area (the local image variations), with the basic assumption that within-crown

brightness variation should not exceed the between-crown variation. Two representative methods have been extensively investigated for this: (a) The "peak-finding" method [68], to detect the local intensity apex as tree tops. To ensure robustness and consistency, this idea can has been combined with image processing techniques such as pre-filtering or smoothing [70], and at multiscale [16]. (b) The contour-based "alley-following" method by Gougeon [37] and others, mainly to delineate the possible crown regions.

We review two exemplar works of this group. Uuttera et al. [84] proposed a process that applies both a smoothing filter and Gaussian-Laplacian filter on images. This process determines the location of a crown by method (a), then uses a region growing method which is similar to method (b), to predict tree crown size. In Gougeon's ITC model [38], image data are first pre-processed to ensure smooth colour variation within a selected region of interest. The tree canopies are then delineated by following the shaded valleys among tree crowns. After determining the position and size of the tree crowns, the tree species signatures are generated and the classification step takes place. After post-processing, the final results are produced. This ITC model was tested on MEIS II sensor data (with about a 30cm/pixel resolution), and in one reported result, 81 percent of the classified crowns coincided with ground-truth obtained by the visual interpretation of experts.

The task of image-based forest inventory is essentially to invert the image formation process: given 2D image observations, estimate the latent scene geometry (3D models and their symbolic interpretation) for the plot of interest. This is an ill-defined problem considering the uncertainties inherent in the image formation process, the lack of adequate observations and correspondences.

Single-view image data limits the capacity of reasoning about the related 3D information considerably. To partially address this limitation, Korpela [51] proposed a stereo method in the context of multiple views for a given scene. It cross-correlated images to detect disparities among these views, by determining the local maxima in the disparity image space for the tree canopies. Unfortunately, this approach made no further attempt to check the consistency with the original image data. It also did not identify the tree species.

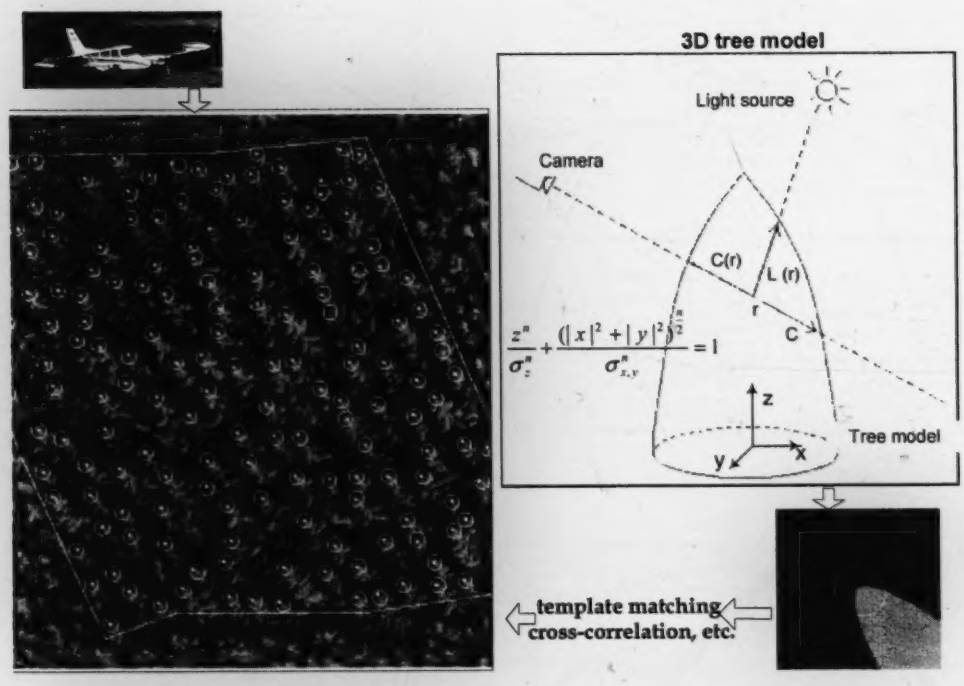


Figure 1.1: This figure shows the basic process-flow of the template-based approaches. After aerial images are taken, templates are generated by rendering the 3D super-ellipsoids into 2D images, and are then used to cross-correlate images to detect optimal matches.

1.2 Template-based Approaches

Instead of directly searching tree tops in the (preprocessed) image data by examining local image convexities, this group of approaches use "3D tree hypothesis". From observations and prior assumptions, a specialized set of 3D models is proposed for each tree species, by considering the contextual knowledge including the camera geometry and scene irradiance (sun-crowns-ground interaction), and by emulating the image generating process. Three-dimensional models are projected to 2D to form a pool of candidate crown templates. then tree apexes are located by cross-correlation of the appropriate template with the image data. This is schematically shown in Fig. 1.1.

The STCI system by Pollock [69] was the first such approach, followed by Larsen et al. [52] and others. Pollock [69] proposed to use the "generalized ellipsoid of revolution" (GER) for 3D tree models

$$\frac{|z|^n}{\sigma_z^n} + \frac{(|x|^2 + |y|^2)^{\frac{n}{2}}}{\sigma_{x,y}^n} = 1. \tag{1.1}$$

This turns out to be a special case of the broader 3D geometric primitive — the super-ellipsoid [87] model which has been adopted in computer-aided design (CAD) [27]

$$|z|^{\frac{2}{n}} + (|x|^{\frac{2}{e}} + |y|^{\frac{2}{e}})^{\frac{e}{n}} = 1. \tag{1.2}$$

Notice Eq. (1.2) [1] results in a normalized 3D tree primitive, and needs to be scaled in the three directions σ_x, σ_y and σ_z respectively in the modelling process.

In details, in the STCI system by Pollock [69], the upper part of a tree crown (so-called "sun crown") is modelled according to Eq. (1.1). By considering the radiometric effects (scene irradiance, camera irradiance, etc.) and camera geometry, the 3D model is ray-traced back to generate a 2D crown template. Tree apexes are found by cross-correlation. The unknown parameters involved are divided into two groups: those that are scene-dependent, and the remainder that are individual crown-dependent. A training procedure is used to decide the scene-dependent parameters via a variant of cross-validation. At run time, after the initial matching via cross-correlation, a pool of candidate tree apexes is detected with associated crown size and species. Then a simulated annealing procedure is used to eliminate the excess false apexes and to estimate the crown dependent parameters. This leads to a consistent interpretation of the tree apexes that are left. Larsen et al. [52] proposed an approach that adopted a simplified strategy to speed up the process. The scene-dependent parameters are largely predetermined. The individual crown-dependent parameters are estimated with a greedy search method.

Since this type of approaches is crucially dependent on the close emulation of the image formation process, they are limited by the capability of how well the synthesized crown templates are matched with the true counterparts. Unfortunately, the variability and non-stationary nature of the image formation process limits the success of this template-based type approaches.

The above methods only have satisfactory results when dealing with certain type of well-structured [2] forestry images under ideal lighting conditions. For example, Larsen et al. [52] reported on a panoramic forest image dataset with 15cm per pixel resolution, recover $\geq$ 91% of the location of the tree apexes. However, as mentioned, such systems are not designed to deal with forestry image data in a more complex context with less than well-defined lighting condition. Further, they were

[1] In the 3D tree generating and fitting module, we adopt the form of Eq. (1.2).

[2] Well-structured here means the trees in the plot are uniformly distributed, well spaced, etc..

designed for single-view, grey-level (panchromatic) imagery data - so having similar disadvantages to the purely image based approaches discussed previously. The estimation of tree volumes could be conducted indirectly via the "taper function" [3] of the estimated 2D tree regions.

1.3 Summary and the Proposed Framework

To address the limitation of the existing approaches, and by following the procedure adopted by expert photo interpreters (see Preface), A new type of Forestry Image Understanding system is proposed. Compared with the image variation approach, it avoids the unrealistic assumption that within-crown variation is smaller than the between-crown variation by way of the proposed supervised segmentation-labelling module. Besides, the multiple-view imagery data are employed to compute the related depth estimation for the segmented tree regions. The task left is that to provide the most plausible scene geometry that are consistent with these multiple (currently we have two) sources, and also consistent with the expertise prior knowledge of the forestry plot of interest. [4] As a result, this proposed approach bypasses the over-simplified assumption of the image formation process inherent in the template-based approach.

In another word, let the image formation process which maps 3D scene geometry to 2D imagery by an unknown function $\mathbf{f}$ as the forward process, and treat the process governed by $\mathbf{f}^{-1}$ as the inverse process. The proposed framework utilizes the interplay between forward and inverse modelling to render the most plausible interpretation of images that will lead to an hypothesized 3D scene model — from a Bayesian perspective.

The proposed framework[5] is schematically shown in Fig.1.2. Given the pre-processed data after the first two modules, namely the labelled regions[6] from the segmenter-annotator module and the depth map from the stereo module, the 3D fitting module iteratively merges and splits the estimated 3D trees, in search for the optimal interpretation of the scene geometry which is consistent with the pre-processed data. The result is a 3D forest derived from the input images. Further

[3] this is a set of non-linear regression functions that can roughly estimate the tree height based on the tree region area from single image. Refer to [45].

[4] These prior knowledge includes for example, the 3D shape and the size for each tree species.

[5] A preliminary version of this framework appeared in [17].

[6] An example of the region label set consists of four labels: aspen, spruce, shadow of both trees and ground (which includes anything else that does not fall into the first three labels).

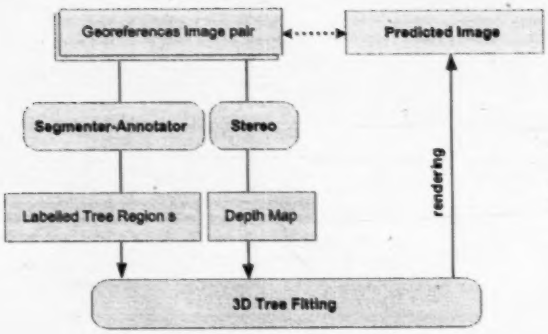


Figure 1.2: Proposed forestry inventory framework. The three sub-processes are the **segmenter-annotator** module to generate labelled regions, the **stereo** module to generate depth map, and the **3D trees fitting** module to produce plausible scene geometry.

rendering into 2D images via the known camera geometry (camera parameters include camera baseline, focal length, field of view, etc.) will lead to the predicted images that are comparable to the input images (as shown in Fig.5.1).

Chapter 2

Related Statistical Models and Methods

In this chapter the statistical methods and algorithms related to the FORIUS system are reviewed. First, the Bayesian paradigm is introduced and posed as an optimization problem. Second, the Expectation Maximization (EM) is reviewed in terms of a bounded optimization technique. Third, to tackle the difficulties encountered while applying the Bayesian approach, various approximate inference methods (Mean Field, Markov Chain Monte Carlo sampling algorithms, etc.) and their relationships are reviewed. Third, as a concrete example, the Hidden Markov Random Field (HMRF) model is introduced, and a systematic way to apply the above-mentioned algorithms onto HMRF model is derived. The content in this chapter reflects our review of the related statistical techniques. Also, we try to provide our own derivations where possible. Most of the results are in line with the existing literature, proving a systematic review of these algorithms. In particular, the analysis of the EM algorithm and that of the HMRF model may shed some light on further investigations.

2.1 Bayesian Analysis

Consider the scenario of complete data $d = (x, y)$ with $x \in \{x_t\}$, $y \in \{y_t\}$ and $t \in [1, T]$, where y_t denotes one sample of observed data and x_t the corresponding latent or unobserved data. T is the number of samples. Assume that data d are generated according to a certain probability function [1] $p(x, y|\theta)$ [2] parameterized by θ. The prior over parameter, $p(\theta|M)$, is another probability function given the model, M. This simplified scenario is of central importance in the thesis since all the three modules of our system can be formulated in terms of its variants. Under the Bayesian paradigm, all unknown quantities are treated as random variables, and the uncertainty over these quantities are represented as probability functions. Fig.2.1 shows its corresponding probabilistic graphical model interpretation,[3] where the left panel shows the general missing data scenario as described so far, with the full probability model taking the form

$$p(x, y, \theta, M) = p(x, y|\theta)p(\theta|M)p(M). \tag{2.1}$$

[1] Here we do not intend to give a rigorous definition on these well-known terms in statistics. Therefore, we simply omit the difference between the probability density function for continuous x and the probability mass function for discrete x. Similarly, when using integrals over, say, random variable x, it is meant to include both the continuous and the discrete cases.

[2] Notice we adopt the generic forms of the quantities x, y, θ. These quantities can be either vector or scalar.

[3] For details of the probabilistic graphical model, we refer the reader to [48].

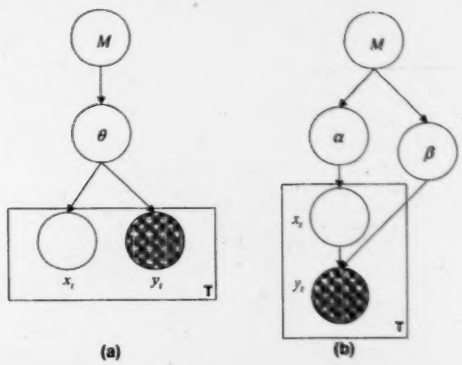


Figure 2.1: This figure shows the probabilistic graphical model interpretation for applying the Bayesian approach to the missing data scenarios (see text for details).

The right panel is a variant of the left, with y generated from x. This leads to the further factorization of the complete data probability function as

$$p(x, y|\theta) = p(y|x, \beta)p(x|\alpha).$$

Here θ is split into two parts, $\theta = (\alpha, \beta)$. To make life easier, we will focus on the generic model shown in the left panel. And it is straightforward to extend to the model in the right panel when necessary.

From Eq. (2.1), an important observation is that the Maximum Likelihood (ML) estimation emphasize on only the first term of the RHS (right hand side), $p(x, y|\theta)$ which produce in frequentist terminology, the "unbiased estimator" — the ML estimator from the Bayesian viewpoint. This ML estimator is a special case of Eq. (2.1) where θ (M) [4] is set to a certain value in the corresponding parameter space with probability 1. This can be interpreted as our *a priori* certainty about both θ. By fixing only the model M, we come to a joint probabilistic model conditioned on M

$$p(x, y, \theta|M) = p(x, y|\theta)p(\theta|M)$$

which is called Generalized Maximum Likelihood (GML) or ML–II in the literature [55] [88]. We will come back to these two point estimation methods [5] later in this section.

[4] The model M has a corresponding hyper-parameter, γ, that parameterizes the prior probability function $p(M) = p(\gamma)$. Later, we interchangeably write $p(\theta|M)$ as $p(\theta|\gamma)$, and $p(M)$ as $p(\gamma)$.

[5] These methods are called point estimation methods since for the function of interest, they focus on reaching the optimal points rather than evaluating also the supporting distribution for these points, as in Bayesian paradigm.

To conclude, the full probability model in Eq. (2.1), as an ideal Bayesian situation, is the generic form which includes ML and GML as special cases. However in practice, it is often intractable to analytically compute Eq. (2.1), and we need to approximate this form via approximation techniques including deterministic methods (variational EM, variational Bayes, [6] etc.) and stochastic sampling techniques (important sampling, Markov Chain Monte Carlo sampling, etc.).

Assume for now that we can analytically compute the full probability model in Eq. (2.1). We write the joint posterior probability function

$$p(x, \theta, M|y) = \frac{p(x, y, \theta, M)}{p(y)}$$

with $p(y) = \int_\gamma \int_\theta \int_x p(x, y, \theta, M) \mathbf{d}x \mathbf{d}\theta \mathbf{d}\gamma$. Thus, we are ready to infer the quantities of interest, namely x, θ and γ. [7] Following Mackay's approach [55], we divide the inference task into three levels of inference:

Level 1 Infer the latent data x

$$p(x|y, \theta, M) = \frac{p(x, y|\theta)p(\theta|M)p(M)}{\int_x p(x, y|\theta)\mathbf{d}x \, p(\theta|M)p(M)} \tag{2.2}$$

$$= \frac{p(x, y|\theta)}{p(y|\theta)} \tag{2.3}$$

where $p(y|\theta) = \int_x p(x, y|\theta)\mathbf{d}x$ is called the evidence of level 1.

Level 2 Infer θ (parameter estimation)

$$p(\theta|y, M) = \frac{\int_x p(x, y|\theta)\mathbf{d}x \, p(\theta|M)p(M)}{\int_\theta \int_x p(x, y|\theta)p(\theta|M)\mathbf{d}x\mathbf{d}\theta \, p(M)} \tag{2.4}$$

$$= \frac{p(y|\theta)p(\theta|M)}{\int_\theta p(y|\theta)p(\theta|M)\mathbf{d}\theta} \tag{2.5}$$

$$= \frac{p(y|\theta)p(\theta|M)}{p(y|M)} \tag{2.6}$$

where $p(y|M) = \int_\theta p(y|\theta)p(\theta|M)\mathbf{d}\theta$ is called the evidence of level 2.

Level 3 Infer M (model choice)

$$p(M|y) = \frac{p(y|M)p(M)}{p(y)} \tag{2.7}$$

where $p(y) = \int_M p(y|M)p(M)\mathbf{d}M$ is called the evidence of level 3.

[6] Variational EM and variational Bayes are deterministic techniques to solve ML and GML, respectively.

[7] The inference of x, θ and γ is also called the inference/parameter estimation/model choice problem in the literature [30], and are unified seamlessly under the Bayesian paradigm. We will interchangeably use these terminologies.

From these three levels of inference, we observe an interesting pattern. Each evidence (i.e. the denominator) of the previous level is employed as a conditional probability function term in the numerator of the current level, and also a term in the integral that forms the evidence of current level. This observation is rather significant, since the log of evidence functions of the first two levels, $\log p(y|\theta)$ and $\log p(y|M)$ are the objective functions to be maximized, by varying the random variable [8] of interest in the next level, for the ML and the GML framework correspondingly. These levels of inference provide a clear and systematic view that unifies, from simple to complex, the two point estimations and the ideal Bayesian framework. Later, we will follow this line in the derivation of variational EM and variational Bayes methods.

One potential question the reader may ask is: what about the maximum *a posteriori* (MAP) [9] compared to the GML framework? The answer is: the MAP turns out to be a special case of GML. In another word, it is a point estimate version of GML after assuming the parameter θ attains its optimal (MAP) solution at a certain value θ^*. To be specific, let us compare the inference of x in the two frameworks

MAP

$$\begin{aligned} p(x|y,\theta^*,M) &= c_0 p(x,\theta^*|y,M) && (2.8)\\ &= c_0 \frac{p(x,\theta^*,y|M)}{\int_\theta \int_x p(x,y,\theta|M)\delta(\theta=\theta^*)\mathbf{d}x\mathbf{d}\theta} && (2.9)\\ &= c_0 \frac{p(x,y|\theta^*)p(\theta^*|M)}{\int_x p(x,y|\theta^*)p(\theta^*|M)\mathbf{d}x} && (2.10)\\ &= c_1 p(x,y|\theta^*)p(\theta^*|M) && (2.11) \end{aligned}$$

GML

$$\begin{aligned} p(x|y,M) &= \int_\theta p(x,\theta|y,M)\mathbf{d}\theta && (2.12)\\ &= c_2 \int_\theta p(x,y|\theta)p(\theta|M)\mathbf{d}\theta && (2.13) \end{aligned}$$

where c_0, c_1 and c_2 are normalizing constants that are irrelevant with the random variable of interest, and can be safely ignored. $\delta(\cdot)$ is the Dirac delta function that attains 1 when $\theta = \theta^*$ and 0 otherwise.

[8] θ and M, respectively.

[9] MAP could be viewed as a penalized or regularized ML.

2.2 EM algorithm and the variants

We review the EM algorithm from several perspectives. Firstly, the EM algorithm is introduced, and the relationship of KL divergence to the auxiliary function is discussed. Secondly, EM is expressed in terms of variational EM, and related to the Fenchel conjugate duality. Thirdly, an extension to variational Bayes is made, based on variational EM. Fourthly, the possibility of upgrading the linear bound to higher-order polynomial is discussed. As one variant of EM, we will also review stochastic EM (SEM), which will serve as a bridge that leads to the sampling techniques in the next section.

2.2.1 Introduction

Let y_t denote one sample of observed data, x_t be the corresponding hidden data, and T be the number of samples. As is shown in the right panel of Fig.2.1, consider complete data $d = (x, y)$ with $x \in \{x_t\}$, $y \in \{y_t\}$ and $t \in [1, T]$. Data d are generated according to a certain distribution function $p(x, y|\theta)$ parameterized by θ. Given the observed $Y = \{y_t\}$[10], the goal is to infer the latent variables $X = \{x_t\}$ and estimate the parameters θ. Expectation Maximization algorithm is especially fit for this situation [25, 61].

2.2.2 The EM Algorithm

By introducing an auxiliary function $q(x)$ with the constraint $\int_x q(x)\mathbf{d}x = 1$, we can formulate:

$$p(y|\theta) = \int_x p(x, y|\theta)\mathbf{d}x = \int_x \frac{p(x, y|\theta)}{q(x)} q(x)\mathbf{d}x$$

according to a variant of Jensen's inequality, the inequality of arithmetic and geometric means, we have

$$p(y|\theta) \geq \prod_x \left(\frac{p(x, y|\theta)}{q(x)}\right)^{q(x)}.$$

Define $L(\theta; y)$ the incomplete data log-likelihood

$$L(\theta; y) = \log p(y|\theta) = \log \int_x p(x, y|\theta) \geq \int_x q(x) \log\left(\frac{p(x, y|\theta)}{q(x)}\right)\mathbf{d}x = \mathcal{L}(\theta, q) \quad (2.14)$$

[10] Which is called the incomplete data, and is in contrast with the complete data d.

where $\mathcal{L}(\theta, q)$ is the lower bound of $L(\theta; y)$ [11] Subtract $\mathcal{L}(\theta, q)$ from $L(\theta; y)$

$$\begin{aligned} L(\theta; y) - \mathcal{L}(\theta, q) &= \log p(y|\theta) - \int_x q(x) \log \frac{p(x, y|\theta)}{q(x)} \mathbf{d}x \\ &= \int_x q(x) \log \frac{q(x)}{p(x|y, \theta)} \mathbf{d}x \\ &= KL(q(x)||p(x|y, \theta)) \end{aligned}$$

where $KL(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence, a measure of the closeness of two distributions. Thus, obtain

$$L(\theta; y) = \mathcal{L}(\theta, q) + KL(q(x)||p(x|y, \theta)) \tag{2.15}$$

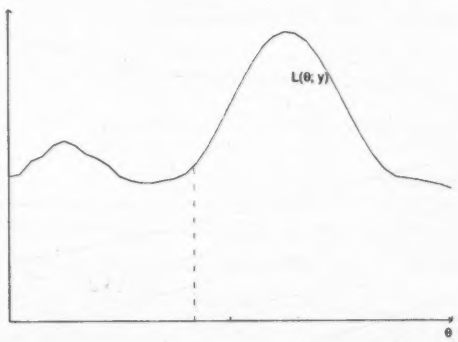


Figure 2.2: The diagram schematically shows, as function of the parameters θ, one log-evidence function $L(\theta; y)$ and its lower bound $\mathcal{L}(\theta, q)$.

As shown in Fig.2.2, given θ, $L(\theta; y)$ is a function of y. By fixing $y = \hat{y}$, EM tries to lower bound $L(\theta; \hat{y})$ by its infimum $\mathcal{L}_{\hat{y}}(\theta, q)$ for $q = q(x)$. The maximum lower bound ($L(\theta; y) = \mathcal{L}(\theta, q)$) is obtained when $q(x)$ reaches its optimal point

$$q(x) = p(x|\hat{y}, \theta). \tag{2.16}$$

In other words, $KL(q(x)||p(x|y, \theta)) = 0$. (Refer to appendix B.2 for the proof.)

To this end, the EM algorithm is interpreted as an alternative maximization method for $\mathcal{L}(q, \theta)$

E step

$$q(x)^* = \arg\max_{q(x)} \mathcal{L}(q(x), \theta), \tag{2.17}$$

[11] $\mathcal{L}(\theta, q)$ sometimes is called the auxiliary function or Q function in the literature.

M step

$$\theta^* = \arg\max_{\theta} \mathcal{L}(\theta, q) \tag{2.18}$$

$$= \arg\max_{\theta} \int_x q(x) \log p(x, y|\theta) \mathbf{d}x. \tag{2.19}$$

Notice that $L(\theta; \hat{y})$ is partitioned into the sum of two terms: a convex term $KL(q||p)$,[12] and a concave term $\mathcal{L}(\theta, q)$. Provided the Hessian of $L(\theta; \hat{y})$ is bounded, we essentially recover the concave-convex procedure (CCCP) [92] in this formulation of EM. Due to the space limit [13], we will not give details toward this direction, readers can instead refer to Yuille et al. [92] for proofs.

2.2.3 Variational EM

In this section, to facilitate the derivation of variational EM, we restrict the distribution function to be within the exponential family.[14] This way, EM can be viewed as alternating lower bounds of the log-partition function. [15]

Complete data case

First, consider a trivial case when we observe the complete data (x, y) with only the parameter θ of the distribution function $p(x, y|\theta)$ unknown. The data are generated from the exponential family distribution

$$p(x, y|\theta) = \exp\{< \theta, \phi(x, y) > -Z(\theta)\} \tag{2.20}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $\phi(x, y)$ is the statistic, and the log-partition function

$$Z(\theta) = \log \int_x \int_y \exp\{< \theta, \phi(x, y) >\} \mathbf{d}y \mathbf{d}x. \tag{2.21}$$

Then the objective function $L(\theta; x, y)$ is

$$L(\theta; x, y) = \log p(x, y|\theta) =< \theta, \phi(x, y) > -Z(\theta). \tag{2.22}$$

[12] The $KL(q||p)$ function is a convex function with respect to q, see [14].

[13] The EM algorithm can be interpreted from many viewpoints which includes the variational approach as we discussed in details here, the CCCP algorithm, the information geometry [4], among others.

[14] Notice that the conventional EM algorithm does not have this restriction, as is clear from the derivation of Eq. (2.17,2.19) in Sec. 2.2.2.

[15] The log-partition function refers to logarithm of the denominator. We will give examples later. Notice the close relationship of certain log-partition function and the log-evidence function.

Since the first term ($< \theta, \phi(x,y) >$) is linear, and the second term $Z(\theta)$ is convex with respect to θ, and $L(\theta; x, y)$ is a concave function, then the (primal) problem becomes

$$\theta = \arg\max_{\theta} L(\theta; x, y). \tag{2.23}$$

Define the Fenchel dual [14] of $Z(\theta)$[16], $Z^*(\mu)$, as

$$Z^*(\mu) = \sup_{\mu}\{< \mu, \theta > -Z(\theta)\},$$

then we also have

$$Z(\theta) = \sup_{\theta}\{< \mu, \theta > -Z^*(\mu)\}. \tag{2.24}$$

Therefore θ can be solved by treating as $\theta = \theta(\mu)$ and finding the critical point of Eq. (2.24). The dual problem of Eq. (2.23) is

$$\mu = \arg\max_{\mu}\{< \theta, \mu > -Z^*(\mu)\}.$$

Incomplete data case

In the incomplete data case y is observed with x and θ unknown. Given the joint distribution function $p(x, y|\theta)$ in Eq. (2.20), we write the posterior probability of x as

$$p(x|y,\theta) = \frac{\exp\{< \theta, \phi(x,y) >\}}{\int_x \exp\{< \theta, \phi(x,y) >\}\mathbf{d}x} = \exp\{< \theta, \phi(x,y) > -Z_y(\theta)\}$$

with the log-partition function

$$Z_y(\theta) = \log \int_x \exp\{< \theta, \phi(x,y) >\}\mathbf{d}x.$$

It is easy to show that both log-partition functions $Z(\theta)$ and $Z_y(\theta)$ are convex [85, 14]. Similar to the complete data case, we also define the Fenchel duals of $Z_y(\theta)$ as

$$Z_y^*(\mu) = \sup_{\theta}\{< \mu, \theta > -Z_y(\theta)\}, \tag{2.25}$$

$$Z_y(\theta) = \sup_{\mu}\{< \mu, \theta > -Z_y^*(\mu)\} \tag{2.26}$$

and we show that (see Eq. (2.30) below)

$$Z_y^*(\mu) = \int_x q(x) \log q(x)\mathbf{d}x.$$

[16] It is well-known that the log-partition function $Z(\theta)$ is a cumulant generating function, which is convex for $\nabla_\theta Z(\theta) = \mu$, the expectation of the sufficient statistics, $\mu = \int_x \int_y p(x, y|\theta)\phi(x, y)\mathbf{d}x\mathbf{d}y$ [85]. Notice that μ is used here as the first-order derivative of the log-evidence function $Z(\theta)$. To simplify the notation, later on μ will be used instead to associate with another log-evidence function $Z_y(\theta)$.

Now we are ready to derive the objective function

$$\begin{aligned} L(\theta; y) &= \log \int_x p(x, y|\theta) \mathbf{d}x && (2.27) \\ &= Z_y(\theta) - Z(\theta) && (2.28) \\ &\geq < \theta, \mu > -Z_y^*(\mu) - Z(\theta) && (2.29) \\ &= \mathcal{L}(\mu, \theta) && (2.30) \\ &= -F(\mu, \theta) && (2.31) \end{aligned}$$

where $F(\mu, \theta)$ is called "free energy" in the literature [64]. Unlike the complete data case, the objective function $L(\theta; y)$ is not concave, since it involves the subtraction of two convex functions. However, the lower bound function $\mathcal{L}(\mu, \theta)$[17] is obviously a concave function since the first term is linear and the second and third terms are convex. This analysis also coincides with what we have shown previously in Eq. (2.15).

To sum up, the optimization problem is

$$\begin{aligned} \sup_\theta L(\theta; y) &\geq \sup_{\mu, \theta} \mathcal{L}(\mu, \theta) && (2.32) \\ &= \sup_{\mu, \theta} \{< \theta, \mu > -Z_y^*(\mu) - Z(\theta)\}. && (2.33) \end{aligned}$$

From Eq. (2.33), we can reformulate the EM algorithm from a variational viewpoint

- E step

$$\mu^* = \arg\max_\mu \{< \mu, \theta > -Z_y^*(\mu)\}, \tag{2.34}$$

- M step

$$\theta^* = \arg\max_\theta \{< \theta, \mu > -Z_y(\theta)\}. \tag{2.35}$$

This recovers exactly Eq. (2.17) and Eq. (2.19) for the exponential family of functions. However, Eq. (2.34) and Eq. (2.35) clearly represent the RHS of both fixed point equations of EM as convex functions, which leads naturally to the CCCP algorithm [92] as conceptually shown in Fig.2.3. In more detail

[17] $\mathcal{L}(\mu, \theta)$ in Eq. (2.30) is equivalent to $\mathcal{L}(\mu, q)$ in Eq. (2.14), see Appendix B.3 for a proof.

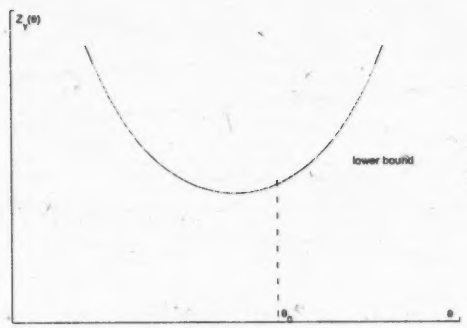


Figure 2.3: This shows the variational EM interpreted as the first-order bounding.

E step Since $\mu = \nabla Z_y(\theta)$ [85], define the inverse mapping

$$\theta_0 = \{\theta, s.t. \nabla Z_y(\theta) = \mu\} = Z_y^{-1}(\mu).$$

Also $Z_y(\theta)$ is convex, thus bounded by the first-order Taylor expansion

$$\begin{aligned} Z_y(\theta) &\geq Z_y(\theta_0) + < (\theta - \theta_0), \nabla Z_y(\theta_0) > \\ &= Z_y(\theta_0) + < (\theta - \theta_0), \mu > \\ &= < \theta, \mu > - (< \theta_0, \mu > - Z_y(\theta_0)). \end{aligned}$$

Assume the maximum lower bound found at the critical point $\theta = \theta_0$, with the gradient $\mu = \nabla Z_y(\theta_0)$. Define the Fenchel dual of $Z_y(\theta)$ at the critical point, $Z_y^*(\mu) =< \mu, \theta_0 > - Z_y(\theta_0)$, and apply this back to the above equation. We recover the E step in Eq. (2.34).

M step As shown in Fig.2.3, we can rewrite $Z_y(\theta)$ by its first order lower bound $< \theta, \mu_1 > + b =< \theta, \mu_1 > - (< \theta_0, \mu_1 > - Z_y(\theta_0))$, and the objective function becomes

$$\inf_\theta \{< \theta, \mu_1 > + b - Z_y(\theta)\} = \inf_\theta \{< \theta, \mu_1 > - Z_y(\theta)\}$$

which recovers the M step in Eq. (2.35).

Compared with the standard EM, the variational EM has both pros and cons

Pros

- It provides a clear link with the optimization techniques, e.g. convex analysis.
- It has a nice primal and dual interpretation (Fenchel duals).

- It provides a solid interpretation for the asymptotic optimality property of approximation in both the E and M steps (Generalized EM), as long as they are toward optimizing the bound.

Cons

- The distributions have to be within the exponential family.

2.2.4 Variational Bayes

By naturally extending the variational framework to have the denominator of the second level of inference as the objective function to be maximized, we obtain the Variational Bayes (VB) method. As mentioned previously in Sec. 2.1, although both are point estimation algorithms, the MAP method is different from the VB method, and in fact could be viewed as a special case of VB.

Let us start from Eq. (2.20) and go further to define the conditional probability for the parameter θ, given the model M

$$p(\theta|M) = \exp\{< \eta, \psi(\theta) > -Z(\eta)\} \tag{2.36}$$

where $\psi(\theta)$ is the statistic of θ, η is the parameter for $\psi(\theta)$, and the log-partition function

$$Z(\eta) = \log \int_{\theta} \exp\{< \eta, \psi(\theta) >\} \mathbf{d}\theta. \tag{2.37}$$

It is clear from the definition of Eq. (2.21) and Eq. (2.37) that both log-partition functions are convex. Thus we write the joint probability function as

$$\begin{aligned} p(x, y, \theta|M) &= p(x, y|\theta)p(\theta|M) & (2.38) \\ &= \exp\{< \theta, \phi(x, y) > + < \eta, \psi(\theta) > -Z(\theta) - Z(\eta)\}. & (2.39) \end{aligned}$$

(In the sequel, it is also helpful to write down the marginal posterior probability for the quantity of interest, as in Appendix B.4.)

The log-evidence is (refer to Sec. 2.1)

$$L(M; y) = \log p(y|M) \tag{2.40}$$

and can be expanded into its integral form and further bounded from below, by

applying Jensen's Inequality

$$
\begin{aligned}
L(M;y) &= \log \int_\theta \int_x p(x,y,\theta|M)\mathbf{d}x\mathbf{d}\theta && (2.41)\\
&\geq \int_\theta q(\theta) \log \frac{\int_x p(x,y,\theta|M)\mathbf{d}x}{q(\theta)}\mathbf{d}\theta && (2.42)\\
&= \mathcal{L}(\eta, q(\theta)) && (2.43)\\
&= \int_\theta q(\theta) \left(\log \int_x q(x) \frac{p(x,y,\theta|M)}{q(x)}\mathbf{d}x - \log q(\theta) \right) \mathbf{d}\theta && (2.44)\\
&\geq \int_\theta q(\theta) \int_x q(x) \log \frac{p(x,y,\theta|M)}{q(x)q(\theta)}\mathbf{d}x\mathbf{d}\theta && (2.45)\\
&= \mathcal{L}((\theta,\eta), q(x)q(\theta)). && (2.46)
\end{aligned}
$$

Further, in Appendix B.4, we establish the equality of $\mathcal{L}(\eta, \mu_\psi)$ and $\mathcal{L}(\eta, q(x))$, and the equality of $\mathcal{L}((\theta,\eta),(\mu_\phi,\mu_\psi))$ and $\mathcal{L}((\theta,\eta), q(x)q(\theta))$.

The essence of variational Bayes is to locate a tighter bound to the function of interest. Let us revisit the log-evidence function in Eq. (2.40) and decompose it into two log-partition functions (Appendix B.4)

$$L(M;y) = Z_y(\eta) - Z(\eta). \tag{2.47}$$

By lower bounding $Z_y(\eta)$, we have the Fenchel duals

$$
\begin{aligned}
Z_y(\eta) &= \sup_{\mu_\psi}\{< \eta, \mu_\psi > - Z_y^*(\mu_\psi)\}\\
Z_y^*(\mu_\psi) &= \sup_{\eta}\{< \mu_\psi, \eta > - Z_y(\eta)\}
\end{aligned}
$$

with, from Appendix B.4

$$Z_y^*(\mu_\psi) = \int_\theta q(\theta) \log \frac{\int_x p(x,y|\theta)\mathbf{d}x}{q(\theta)}\mathbf{d}\theta. \tag{2.48}$$

Then we can derive the EM-style algorithm for VB as

E step $\mu_\psi^* = \arg\max_{\mu_\psi}\{< \mu_\psi, \eta > - Z_y^*(\mu_\psi)\}$

M step $\eta^* = \arg\max_{\eta}\{< \eta, \mu_\psi > - Z(\eta)\}$.

Notice, however, $Z_y^*(\mu_\psi)$ is not a convex function as is clear from Eq. (2.48). We proceed by further lower-bounding it, and after some derivation (see Appendix B.4) we have

$$
\begin{aligned}
Z_y^*(\mu_\psi) &= \int_\theta q(\theta)\log q(\theta)\mathbf{d}\theta - \int_\theta L(\theta, y)\mathbf{d}\theta && (2.49)\\
&\geq \int_\theta q(\theta)\log q(\theta)\mathbf{d}\theta - \int_\theta q(\theta)(< \theta, \mu_\phi > - Z_y^*(\mu_\phi) - z(\theta))\mathbf{d}\theta. && (2.50)
\end{aligned}
$$

Notice the second term of the RHS of Eq. (2.50) shows Minka's idea of lower bounding [62]. By substituting Eq. (2.50) into Eq. (2.48) and letting $\mu_\theta = E_{q(\theta)}[\theta]$, we have

$$\begin{aligned} L(M;y) &\geq < \eta, \mu_\psi > -Z(\eta) - \int_\theta q(\theta)\log q(\theta)\mathbf{d}\theta + \int_\theta (< \theta, \mu_\phi > -Z_y^*(\mu_\phi) - Z(\theta))\mathbf{d}\theta \\ &= < \eta, \mu_\psi > -Z(\eta) - Z_y^*(\mu_\psi) + < \mu_\theta, \mu_\phi > - \int_\theta (\log q(\theta) + Z(\theta))\mathbf{d}\theta \\ &= < \eta, \mu_\psi > + < \mu_\theta, \mu_\phi > -Z(\eta) - Z_y^*(\mu_\phi) - \int_\theta q(\theta)(\log q(\theta) + Z(\theta))\mathbf{d}\theta \\ &= \mathcal{L}((\theta,\eta),(\mu_\phi,\mu_\psi)). \end{aligned}$$

Notice $\mathcal{L}(\eta, \mu_\psi)$ is not a concave function, but the lower-bounding of it, $\mathcal{L}((\theta,\eta),(\mu_\phi,\mu_\psi))$ is a concave function.

Again, we come to a EM-type algorithm, as

Update μ_ϕ

$$\mu_\phi^* = \arg\max_{\mu_\phi}\{< \mu_\theta, \mu_\phi > -Z_y^*(\mu_\phi)\}, \tag{2.51}$$

Update θ, μ_ψ

$$\theta^* = \arg\max_\theta\{< \mu_\theta, \mu_\phi > - \int_\theta q(\theta)(\log q(\theta) + Z(\theta))\mathbf{d}\theta\}$$

$$\mu_\psi^* = \arg\max_{\mu_\psi}\{< \mu_\psi, \eta > + < \mu_\theta, \mu_\phi > - \int_\theta q(\theta)(\log q(\theta) + Z(\theta))\mathbf{d}\theta\}, \tag{2.52}$$

Update η

$$\eta^* = \arg\max_\eta\{< \mu_\eta, \mu_\psi > -Z(\eta)\}.$$

And after some derivation (Appendix B.4), we have the closed form equations

$$p(x|y,\theta) = \frac{\exp\{\int_\theta q(\theta)\log p(x,y|\theta)\mathbf{d}\theta\}}{\int_x \exp\{\int_\theta q(\theta)\log p(x,y|\theta)\mathbf{d}\theta\mathbf{d}x} \tag{2.53}$$

$$p(\theta|x,y,M) = \frac{\exp\{p(\theta|M)\int_x q(x)\log p(x,y|\theta)\mathbf{d}x\}}{\int_\theta \exp\{p(\theta|M)\int_x q(x)\log p(x,y|\theta)\mathbf{d}x\}\mathbf{d}\theta}. \tag{2.54}$$

Since in the EM-type algorithm, Eq. (2.51) and Eq. (2.52) result in meaningful closed form updating equations Eq. (2.53) and Eq. (2.54). While the other two equations only recover the mean parameter constraints, we can simply drop the latter two equations and have the following EM-type algorithm

E step Update $q(x)$ by Eq. (2.53), and

M step Update $q(\theta)$ by Eq. (2.54).

This is he Variational Bayes (VB) algorithm advocated by Beal et al. [8]. As discussed, this algorithm is essentially equivalent to the generalized maximum likelihood (GML) method. It is not a "true" Bayesian algorithm since instead of approximating the posterior distribution of the model of interest $p(M|y)$, it does the point estimation by lower bounding the log-evidence function $\log p(y|M)$. It is clear that by observing Eq. (2.7) and assuming equal prior $p(M)$ over the model space in the simplified scenario, we have

$$p(M|y) = p(y|M).$$

The VB (or GML) method is simply a maximum likelihood (ML) estimation of $p(M|y)$. Thus, as the model becomes complicated, leading to multi-modal in high dimensional space, the VB (GML) method, which inherits the same properties as a ML-type algorithm, can fail to capture the interesting density regions by getting stuck at some critical points (local optimal or saddle points).

2.2.5 Upgrading the Bound on EM

In this section, we discuss possibilities to upgrade the first order lower bound of the standard EM algorithm to higher order polynomial bounds. First, we review the Taylor expansion of a function to establish the relationship of first-order Taylor expansion of convex function and the Fenchel dual. We show that the variational EM is exactly based on the first-order Taylor expansion of the log-partition function. Second, we further study the Taylor expansion of the exponential function and link it with the tighter bounds reported in the literature [53], and demonstrate the possibility to represent the variational EM on the basis of Taylor expansion of the exponential function. As a by-product, we discover a new set of Fenchel duals different from those discussed in the previous section (Sec. 2.2.3).

The Taylor Expansion and the Fenchel Dual

A function $f(x)$ can be approximated by its Taylor series from a given point. Taken a real function $f(x) : \mathbf{R} \mapsto \mathbf{R}$, expand it from a point x_0, as

$$f(x) \approx f(x_0) + f^{'}(x_0)(x - x_0) + f^{''}(x_0)\frac{(x - x_0)^2}{2!} + \cdots + f^{(n)}(x_0)\frac{(x - x_0)^n}{n!}$$

with equality holding when $n \to \infty$. In particular, for a convex function $f(x)$, we know that at any point x_0, the first order Taylor expansion of $f(x)$ is always a lower

bound

$$f(x) \geq f(x_0) + < \nabla f(x_0), (x - x_0) >$$

with equality holding when $x = x_0$. Let $\lambda = \nabla f(x_0)$,

$$\begin{aligned} f(x) &\geq f(x_0) + < (x - x_0), \lambda > \\ &= < x, \lambda > - (< x_0, \lambda > - f(x_0)). \end{aligned}$$

Now define the Fenchel duals as

$$f^*(\lambda) = \sup_x \{< \lambda, x > - f(x)\}$$

with equality holding at $x = x_0$, and

$$f(x) = \sup_\lambda \{< x, \lambda > - f^*(\lambda)\}.$$

As mentioned earlier, the variational EM we derived in Sec. 2.2.3 is based on the first-order Taylor expansion of the log-partition function, as evident from Eq. (2.25)and Eq. (2.26).

The Fenchel duals based on the exponential function

The inequality of exponential function, based on its first order Taylor expansion, is

$$\begin{aligned} f(E_p) = \exp\{E_p\} &\geq \exp\{E_q\} + < \nabla \exp\{E_q\}, (E_p - E_q) > & (2.55) \\ &\geq \exp\{E_q\}(1 + E_p - E_q). & (2.56) \end{aligned}$$

From Eq. (2.20), we have the distribution function of complete data, as LHS

$$f(E_p) = p(x, y|\theta) = \exp\{< \theta, \phi(x, y) > - Z(\theta)\}.$$

Thus we can define $E_p = \log p(x, y|\theta) =< \theta, \phi(x, y) > - Z(\theta)$, and define $q(x) = \exp\{E_q\}$, and substitute this back to Eq. (2.56). Again we come to the Fenchel duals as

$$f^*(q) = \sup_{E_p} \{< q, E_p > - f(E_p)\}$$

with equality holding at $E_p = E_q$, where $f^*(q) = -q(x) - q(x) \log q(x)$, and

$$f(E_p) = p(x, y|\theta) = \sup_q \{< E_p, q > - f^*(q)\}. \quad (2.57)$$

The lower bound of the objective function of EM $\log p(y|\theta)$

$$\begin{aligned}\log p(y|\theta) &= \log \int_x p(x,y|\theta)\mathbf{d}x \\ &\geq \log \int_x (< q(x), E_p(x) > - f^*(q(x))\})\mathbf{d}x.\end{aligned}$$

Notice that this is in line with lower bound integration method by Minka [62].

It is interesting to note that the Fenchel dual variables are $\log p(x,y|\theta) = \{< \theta, \phi(x,y) > - Z(\theta)\}$ and $q(x)$, which are different from what we derived previously in variational EM (Sec. 2.2.3) where the duals are θ and μ. It is easy to prove that both duals have the same maximizer, but this latter form is more general since, a) it has been used to derive the mean field (MF) and generalized mean field (GMF) approximation algorithms as in [90], and b) it can be further augmented to higher-order polynomials, as was proposed in [53], where Eq. (2.56) is the basis function to construct the lower bound by such higher-order polynomials. It can be proved that when applied into Eq. (2.57), versus the Jensen's inequality applied in Eq. (2.33), they have the same maximizer. However, with these higher-order polynomial bounds, the duals obtained are beyond the Fenchel duals which corresponds only to the first-order variational bound.

2.2.6 Stochastic EM

EM algorithm is suitable in situations involving missing data and unknown parameters. The E step can be viewed as computing expectation of the sufficient statistics which in turn, can be regarded as a integral over all possible configurations of the latent variable x to form the complete data (x,y). However, for cases where enumeration over x is intractable, as in our 3D tree fitting module in Chapter 5.5, we have to resort to a stochastic version in the E step. There are many of these stochastic-type EM algorithms in the literature. Celeux et al. [19] provides a good survey on this topic. Here we only discuss one variant of the so-called "SEM" algorithm in [19], and is also termed as "Monte Carlo EM" in [86]. Basically the SEM algorithm is as follows.

Sampling step Given a current value of θ, sampling S *iid* samples $\{x^{(s)}; s \in [1,S]\}$ from the posterior probability function $q(x) = p(x|y,\theta)$. We form the pseudo-complete samples $\{(x^{(s)}, y^{(s)}); s \in [1,S]\}$ which represent the sampling approx-

imation of the expectation [18] $E_{q(x)}[\phi(x,y)] = \int_x q(x)\phi(x,y)\mathbf{d}x$, with $\phi(x,y)$ the function as integrand.

M step Maximize the lower bound $\mathcal{L}(\mu,\theta)$ [19] by varying θ values.

To avoid repetition, we put off the details of the implementation to Chapter 5.5. Instead we will only provide some intuitions about the asymptotic properties of the sampling process in the E step. For a rigorous proofs of the robustness and asymptotic convergence properties of SEM, the readers can refer to Nielsen [66] for details.

In the sampling step (E step), by Monte Carlo approximation to the distribution of interest $q(x) = (1/S)\sum_s \delta(x - x^{(s)})$, we are performing the approximation of the expectation

$$\begin{aligned} E_{q(x)}[\phi(x,y)] &= \int_x q(x)\phi(x,y)\mathbf{d}x \\ &\approx \int_x \phi(x,y)\frac{1}{S}\sum_s \delta(x - x^(s))\mathbf{d}x \\ &= \frac{1}{S}\sum_s \phi(x^{(s)},y). \end{aligned}$$

Obviously this approximation is an unbiased estimation of the true value as $s \to \infty$

$$E_{q(x)}[\phi(x,y)] = E\left[\frac{1}{S}\sum_s \phi(x^{(s)},y)\right].$$

Assuming $\hat{\sigma}_S^2$ the variance of the Monte Carlo estimator, σ^2 the variance of the distribution of $q(x)$, we can show that the variance of the sampling approximation as (see Appendix B.1 for the derivation.)

$$\hat{\sigma}_S^2 = \frac{1}{S}\sigma^2. \tag{2.58}$$

From this derivation, it is obvious that the variance of the approximation is independent from the dimensionality of the function under approximation, and will asymptotically decrease with the scale $\frac{1}{S}\sigma^2$ by increasing the size of the sample population. Thus we can approximate the integral of interest up to arbitrary precision by drawing samples of sufficiently large population size. This observation also provide a basis for more advanced and sophisticated sampling techniques, as we will review later.

[18] In connection to the EM algorithms we previously derived, this expectation which is also called the empirical expectation, and is forced to equal the mean parameters, that is, $E_{q(x)}[\phi(x,y)] = (\mu,\sigma)$ for the Gauss pdf.

[19] Refer to Eq. (2.19).

2.3 The Sampling Algorithms

There are a spectrum of approximation algorithms that we attempt to cover in this chapter. As mentioned earlier, these approximation techniques come into play when the function of interest, which typically involves integrations over unknown functions, [20] are analytically intractable. These algorithms are of crucial importance to study, since in practice, there are only a handful of models that are analytically tractable. [21]

Roughly speaking, these algorithms can be partitioned into two regimes, the deterministic algorithms and the sampling algorithms. The deterministic approximation algorithms include: variational EM (VEM) (Sec. 2.2.3); variational Bayes (VB) (Sec. 2.2.4) that are lower-bound approximations to the log-evidence function; Loopy Belief Propagation (LBP) (4.2.1) to approximate the Bethe free energy which itself is an approximation to the log-evidence function [91]; Laplace approximation [55] to approximate the single peak (mode) of certain posterior probabilistic function with second-order Taylor expansion, and so on. In this section, we focus on the algorithms that fall in the other group, the sampling algorithms. In the following paragraphs, we try to provide an intuitive tour of those sampling techniques and their properties. For a more detailed or rigid description, the readers can refer to [63] for a comprehensive tutorial, and refer to Gilks et al. [34] for many practical techniques and successful applications.

We focus on those recent developed sampling algorithms that are relevant or potentially promising to our proposed work. Therefore we skip those otherwise fundamental sampling algorithms in the literature [76], such as the simulated annealing algorithm, particle filters, and so on. Still, we manage to provide pointers to the related literature.

2.3.1 Direct Sampling

Assume the probability function of interest takes the form $p(x) = f(x)/Z_f$ where $f(x)$ is an un-normalized function of $p(x)$ with the normalization constant Z_f, and x the random variable, being either a scalar or a vector. Direct sampling, drawing

[20] An example could be to calculate the posterior probability in Eq. (2.13), where we need to calculate the integral of $p(\theta|M)$ which is another unknown posterior probability function.

[21] Take the undirected graphical model (UGM) for example. The tractable models includes the singly-connected (tree-structured) UGM and those that can be easily converted to junction trees [22]; On the contrary, most UGMs are analytically intractable. The readers should consult Jordan et al. [48] for details.

samples directly from $p(x)$, is the simplest form of the sampling algorithms, with its asymptotic convergence properties discussed in Sec. 2.2.6 when we described the SEM algorithm. However, in many situations, we can only afford to draw samples from a proposal probabilistic function $q(x)$, and providing we can evaluate the value via $f(x)$ [23], then our goal is to make sure that the sampled empirical distribution is as close to the true distribution $p(x)$ as possible.

2.3.2 Importance Sampling and Rejection Sampling

In accordance with $p(x)$, assume the proposal probabilistic function is also of the form $q(x) = g(x)/Z_g$. Let $w(x) = f(x)/g(x)$ be called the importance weight. To compute the expectation of a statistic function $\phi(x)$ over $p(x)$, we have

$$\begin{aligned} E[\phi(x)] &= \int_x p(x)\phi(x)\mathbf{d}x \\ &= \frac{Z_g}{Z_f}\int_x q(x)w(x)\phi(x)\mathbf{d}x \\ &\approx \frac{Z_g}{Z_f}\frac{1}{S}\sum_s w(x^{(s)})\phi(x^{(s)}) \\ &= \frac{\sum_s w(x^{(s)})\phi(x^{(s)})}{\sum_s w(x^{(s)})} \\ &= \hat{E}_S[\phi(x)]. \end{aligned}$$

In other words, to approximate this expectation of $p(x)$, we draw S samples from $q(x)$ and approximate with a convex linear combination of the samples. This sampling method is called importance sampling.

Assume $f(x)$ can be upper-bounded by $Mg(x)$ with M as a constant. A similar method called the rejection sampling, works by:

a) drawing S samples from $q(x)$, with each sample $x^{(s)}$ associated with a value $u(s)$ drawn from uniform distribution $U[0,1]$

b) for each sample $u(s)$, accept it only when

$$u < \frac{p(x^{(s)})}{Mq(x^{(s)})}.$$

Both methods work well when $g(x)$ is a close match of $f(x)$, and fails if $g(x)$ is rather dissimilar with $f(x)$ or worse, when the random variable x is of high dimensionality, which is also called "curse of dimensionality". This observation leads us to

[23] It is typically much harder to compute the normalizing constant Z_p. Mackay's tutorial on Monte Carlo methods [56] gives detailed accounts of the difficulties of drawing samples, or even evaluating the function value directly from $p(x)$.

consider using dependent samples which might result in a tighter approximation to the true density function $f(x)$, instead of stick to independent samples as we do in both previous methods. This is the Markov Chain Monte Carlo (MCMC) sampling technique.

2.3.3 Markov Chain Monte Carlo (MCMC)

As mentioned, when the distribution of interest is of high dimensionality or of a rather complex shape, [24] the independent sampling methods (importance sampling, rejection sampling) may fail miserably. Thus we have to consider more complicated, MCMC techniques which draw samples from a Markov chain. MCMC samplers have many attractive properties. For example, they allow the initial proposal distribution from which we draw the first sample $q(x^{(1)})$ to be far from $p(x)$ the true distribution, and still guarantee final convergence to $p(x)$. The MCMC sampler works as to

(a) draw a sample $x^{(s+1)}$ from the proposal distribution $q(x^{(s+1)}|x^{(s)})$, also draw a random value $\mu \sim U[0,1]$,

(b) accept the sample $x^{(s+1)}$ only when $\mu < A(x^{(s)}, x^{(s+1)})$, where $A(x^{(s)}, x^{(s+1)})$ is called the acceptance probability,

(c) increase s by 1, and repeat (a) and (b) until convergence.

Most importantly, in order to possess the aforementioned nice properties, by assuming the admissible sampling space for x from a countable set Ω, there are three basic principles to be followed when one creates a new MCMC sampler.

Irreducibility Given the current sample $x^{(s)}$ is at a state in the Markov chain, [25] there are always positive probabilities to visit any state $x \in \Omega$. In the multi-mode scenario, this property allows the next sample $x^{(s+1)}$ opportunity to come across lower-probability zones connecting the modes.

Aperiodicity (or recurrence) The chain cannot visit the states in cycles. This allows the stochastic or random-walk behavior of the samplers.

[24] For example, $p(x)$ is multi-modal with near-zero probabilities in-between the modes.

[25] To avoid confusion, we call the sampled sequence $(x^{(1)}, \ldots, x^{(S)})$ the Markov chain or the chain. Also we use kernel matrix $K(x^{(s+1)}|x^{(s)})$ to represent both the transition matrix for discrete space of x and the transition kernel for continuous space of x, where x can be a scalar or a vector.

Invariance (or convergence) Define $p(x)$ as the invariant distribution of the transition kernel $K(x^{(s+1)}|x^{(s)})$. [26] The chain sampled from $K(x^{(s+1)}|x^{(s)})$ converges to the invariance distribution $p(x)$ in law. That is to mean

$$p(x^{(s+1)}) = \int_{x^{(s)}} K(x^{(s+1)}|x^{(s)})p(x^{(s)})\mathbf{d}x^{(s)}. \tag{2.59}$$

These principles form the basic ingredients when constructing a new MCMC sampler. To make the design task simpler, there is a stronger condition which is a sufficient (but not necessary) condition of Eq. (2.59), called the detailed balance [34], as

$$K(x^{(s+1)}|x^{(s)})p(x^{(s)}) = K(x^{(s)}|x^{(s+1)})p(x^{(s)}). \tag{2.60}$$

We can easily prove that Eq. (2.60) leads to Eq. (2.59) by integrating over $x^{(s)}$ on both sides of Eq. (2.60).

In the following we describe several popular MCMC samplers. Due to space limit, we skip the discussion of other samplers which include: the sequential Monte Carlo (or particle filter) [26]; the auxiliary methods to speed up the sampling which includes the Hybrid Monte Carlo sampling (Swendson-Wang dynamic [80]); the Slice sampling [65]; and the samplers that combine the sub-optimal deterministic algorithms with MCMC. [27] Also for the practical issues such as determining when the chain starts to mixing, and deciding if the chain is mixing well, etc., please consult Gilks et al. [34].

Metroplis-Hastings Sampler

As the cornerstone of the MCMC sampling technique, the Metroplis-Hastings (M–H) sampler is extensively used and most MCMC samplers could be understood as variants of the M–H sampler. The basic M–H algorithm works as follows:

1. $s = 0$, initiate the first sample $x^{(s)}$,
2. For $s = 0$ to $S - 1$
   - Sample $\mu \sim U[0, 1]$, and sample $x^* \sim q(x^*|x^{(s)})$
   - Compute $A(x^{(s)}, x^*) = \min\left\{1, \frac{q(x^{(s)}|x^*)p(x^*)}{q(x^*|x^{(s)})p(x^{(s)})}\right\}$

[26] It is easy to prove for x discrete, $p(x)$ is the eigen-vector of the transition kernel $K(x^{(s+1)}|x^{(s)})$ with eigen-value equal to 1.

[27] For example, the Rao-Blackewellization scheme allows the combination of exact inference of a portion of random variables while sampling the rest [18]. Another example is the variational MCMC while combines the deterministic optimization algorithm with MCMC sampling [28].

- if $\mu < A(x^{(s)}, x^*)$

$$x^{(s+1)} = x^*$$

else

$$x^{(s+1)} = x^{(s)}.$$

It is not hard to prove that this algorithmic construction ensures the satisfaction of the detailed balance condition (see Appendix B.5).

There are many special cases of M–H sampler, such as independent sampler [34] where $q(x^*|x^{(s)}) = q(x^*)$, and the Metropolis sampler [34] where $q(x^*|x^{(s)}) = q(x^{(s)}|x^*)$. In practice, however, it turns out that the proposal distribution plays a crucial role in implementing a successful sampler. For example, in the multi-modes scenario, too narrow a proposal distribution would limit the accessibility to other modes, while too broad a proposal would lead to high rejection rate and, as a result, the sampled chain would be highly correlated.

Cycles of MCMC Kernels, Gibbs Sampling

Up until now, we treat the random variable x as a unit when operating the MCMC machinery. It turns out to be beneficial in breaking x into a set of components $x = (x_1, \ldots, x_N)$, and updating one component x_i while fixing the remainder $x_{\setminus i}$, in each instance. This method is called Cycles of MCMC kernels [28] or block-update [34], in the literature. Basically, it works by sampling x^* with proposal probability

$$q_i(x^*|x^{(s)}) = q(x_i^*|x_i^{(s)}, x_{\setminus i}^{(s+1)}),$$

and accepting it with probability

$$A_i(x^{(s)}, x^*) = \min\left\{1, \frac{p(x_i^*|x_{\setminus i}^*)q_i(x^{(s)}|x^*)}{p(x_i^{(s)}|x_{\setminus i}^{(s)})q_i(x^*|x^{(s)})}\right\}$$

where $p(x_i^*|x_{\setminus i}^*)$ and $p(x_i^{(s)}|x_{\setminus i}^{(s)})$ are the invariant distribution for the component of interest, x_i. Gibbs sampling, proposed by Geman et al. [32] for inference of Markov random field (MRF), [28] is a special case of this approach where each site is visited in sequence and its value is sampled (and is accepted with probability 1.) to mimic its conditional invariant density $p(x_i|x_{\setminus i})$. We can easily prove this construction obeys the detailed balance condition in a way similar to the Metroplis-Hastings

[28] Refer to Sec. 2.4 for the Hidden Markov Random Field.

sampler. Further, it can also be proved that the Gibbs sampler is a special case of the Metroplis-Hastings Sampler. In particular, the Iterative Conditional Mode (ICM) [10] method is a deterministic variant of the Gibbs sampler. Given $q_i(x^*|x^{(s)})$ which approximates the marginal posterior distribution for site i, ICM reaches the local optimal position while Gibbs sampling draw a set of samples to capture the whole distribution.

2.3.4 Model Selection with MCMC

One important property of Bayesian methodology is to gracefully unify the otherwise distinct statistical tasks into the inference framework [30]. These tasks include the inference of latent random variables, the parameter estimation problem, and the model selection problem. Therefore, rather than other statistical model selection approaches (Neyman-Pearson likelihood ratio test, cross-validation, Bayes factor, etc.), we discuss the trans-dimensional MCMC model selection approaches that are most relevant to our proposed work in Chapter 5.5.

Trans-Dimensional MCMC

Godsill [36] proposed a "composite model space" approach that contains existing trans-dimensional MCMC approaches as special cases. We adopt this "composite model space" as the starting point.

Let $\Theta = \{\theta_1, \cdots, \theta_N\}$ be the collection of parameters over the model space $\mathcal{K} = \{1, \cdots, K\}$ under consideration. Let $\mathcal{I}(\mathcal{K}) = \{\mathcal{I}(1), \cdots, \mathcal{I}(K)\}$ be the index set, with each element $\mathcal{I}(k) = \{i_1(k), \cdots, i_{l(k)}(k)\}$ containing the indices for these $l(k)$ parameters of model k. Thus, $\theta_{\mathcal{I}(k)} = \{\theta_i; i \in \mathcal{I}(k)\}$ is the set of parameters related to model k. Further, define the solution space of the joint probability function $p(k, \theta, y|M)$ [29] as a product-space representation:

$$(\mathcal{K}, \Theta) = \mathcal{K} \times \prod_{i=1}^{N} \theta_i, \tag{2.61}$$

and define

$$p(k, \theta, y|M) = p(y|k, \theta_{\mathcal{I}(k)}) p(\theta_{\mathcal{I}(k)}|k) p(\theta_{\backslash \mathcal{I}(k)}|\theta_{\mathcal{I}(k)}, k) p(k|M) \tag{2.62}$$

where $\theta_{\backslash \mathcal{I}(k)} = \Theta \backslash \theta_{\mathcal{I}(k)}$ denotes the set of parameters that are not considered in model k. This "composite model space" formalism contains, as special cases, the

[29] For simplicity, here we assume the the complete data are observed, and denoted as y.

jump-diffusion approach (JD–MCMC), the reversible-jump approach (RJ–MCMC), and the data-driven approach (DD–MCMC), among others.

The JD–MCMC approach [40] comprises two possible type of moves: the trans-model jumps and the within-model diffusion which is guided by a Langevin stochastic equation for modelling brownian motion. The RJ–MCMC approach generalizes to unify those moves under the Metroplis-Hastings framework, as cycles of MCMC kernels. Due to the complication of this approach which was derived based on the measure theory, we omit the theoretical analysis of the detailed balance condition, the asymptotical convergence property and so on, and give a concise introduction instead.

By observing Eq. (2.62), $p(\theta_{\backslash \mathcal{I}(k)}|\theta_{\mathcal{I}(k)}, k)$ could be assigned to any proper distribution without affecting the marginal $p(k, \theta_{\mathcal{I}(k)}|y, M)$. Ideally, this distribution should be approximated as close as possible to the invariate distribution which is typically non-uniform. This, however, requires an extra amount of computation or storage. RJ–MCMC simplify this consideration by dropping this term out of Eq. (2.62), assuming it as a uniform distribution. This way, the solution space is reduced to a union-space

$$\Omega = (\mathcal{K}, \Theta) = \bigcup_{k=1}^{K} (k, \theta_{\mathcal{I}(k)}). \tag{2.63}$$

Correspondingly, the joint pdf becomes

$$p(k, \theta, y|M) = p(y|k, \theta_{\mathcal{I}(k)}) p(\theta_{\mathcal{I}(k)}|k) p(k|M). \tag{2.64}$$

As an example of the RJ–MCMC sampling scheme, we show the proposed algorithm [30] in the 3D tree fitting module in Sec. 5.3.2.

To enable a reversible jump from a model m with dimension $d_m = |x_m|$[31] to another model m' with dimension $d_{m'}$, the basic idea behind RJ–MCMC is to match two models up to the same dimensionality. This is done by augmenting the variables x_m with auxiliary variables μ_m, to form (x_m, μ_m), and correspondingly, to form $(x_{m'}, \mu_{m'})$ with auxiliary variables $\mu_{m'}$. Let the dimensions of μ_m and $\mu_{m'}$ be r_m and $r_{m'}$, respectively. We have

$$d_m + \mu_m = d_{m'} + \mu_{m'}.$$

[30] Notice that our approach is different from RJ–MCMC inside the jump moves.

[31] To ease the notation, we adopt x_m to refer to the parameter set for model m $\theta_{\mathcal{I}(m)}$. This change of notation lasts for the rest of this section.

To satisfy the detailed balance condition, when proposing a jump move from x_m to $x_{m'}$ ($|x_m| \neq |x_{m'}|$), the RJ–MCMC sampler is constructed as to

- choose auxiliary variables μ_m and $\mu_{m'}$ by sampling from auxiliary proposal distribution $q(\mu_m)$, $q(\mu_{m'})$
- accept this move with acceptance probability

$$A_{m \to m'} \min\{1, \frac{p(m', x_{m'})}{p(m, x_m)} \frac{q(m|m')}{q(m'|m)} \frac{q(\mu_{m'})}{q(\mu_m)} J_{m \to m'}\} \tag{2.65}$$

where the term $J_{m \to m'}$ is the Jacobian for the change of variables from (m, μ_m) to $(m', \mu_{m'})$ [49].

In practice, it turns out that the proposal distributions in the jump moves are critical to the efficiency of the convergence of the chain. As observed by Zhu et al. [82], due to the inherent randomness introduced in the auxiliary variables, between each jump, often the RJ-MCMC samples spend a long time on portions of the distribution space that are less plausible. To reduce the amount of waiting time, Zhu et al. [82] proposed to incorporate the data-driven heuristics into the proposal probabilities $q(\cdot)$.

DD–MCMC is quite similar to RJ–MCMC in that it also adopts the union-space formalism. The major differences lie in that

- DD–MCMC incorporates the sub-optimal algorithms to form the proposal probabilities, [32]
- By cleverly arranging the computation of the jump back and forth, the Jacobian terms in Eq. (2.65) which are cumbersome to compute in practice, are cancelled out.

In practice, the DD–MCMC approach has successes in applications such as image segmentation and texture analysis [82] [81].

2.3.5 The Roles of MCMC and Deterministic Algorithms in Approximation

As mentioned earlier, there is a spectrum of approximation techniques for solving otherwise intractable function approximation problems (the integration of unknown

[32] For example, in the image segmentation application [82], an ensemble of existing edge-detection and data-clustering algorithms are employed to produce sub-optimal candidate image regions, and these solutions are combined by associated weights to form a proposal probability function.

functions, constraint and unconstraint optimization problems, etc.). At one end of this spectrum, there are these deterministic methods which contain, besides the standard optimization techniques, [33] those deterministic algorithms that are closely related to statistical inference and estimation, including EM, variational methods, CCCP and convex analysis [14], LBP, MF(GMF), etc. Meanwhile, at the other extreme of this spectrum, lies a set of sampling algorithms, as discussed above.

Generally speaking, the deterministic techniques, as point estimation methods, enjoy the benefit of fast computation. In particular, the variational algorithms including EM and CCCP can guarantee optimality (critical point) of the function of interest, especially when the function has nice properties (e.g., convexity, continuity) or low dimensionality. On the contrary, the sampling algorithms are rather computationally intensive, and often special care needs to be taken to ensure the chain converges to the invariant distribution, which may require experience and expertise beyond the algorithms, per se.

On the other side, deterministic methods easily get trapped in local optima or saddle points. With the growth of the complexity of the models, the function of interest may be highly multi-modal in high dimensionalities. The sampling approach turns out to be preferred. In practice, under the powerful Bayesian paradigm, many successes ([34], [39], [41], [82], [63]) have been achieved with highly complex models that prevent the deterministic approach from being applicable. Further, the sampling approach often provides, besides an optimal solution, an estimation of the posterior distribution for the unknown quantities. This fits naturally with the Bayesian philosophy of analyzing the uncertainty toward unknown quantities of interest.

This is, however, by no mean to say that the two regimes are clearly separated. It turns out that they are complementary to each other in most cases. In some situations, a sensible strategy is to employ them jointly to have the benefits of both approaches. An example is the DD–MCMC algorithm in Sec. 2.3.4 that adopts a set of deterministic methods to find the sub-optimal local solutions, which are then combined as the proposal probabilities for the JD–MCMC machinery to choose the optimal solution in the optimal solution state-space. In our work, we use a similar idea in the stereo vision module in Chapter 4.4, to combine the deterministic

[33] The standard optimization techniques include gradient descent, Newton-Rhapson method, linear programming, etc., refer to classical books on numerical methods and optimization.[71]

LBP algorithm for approximate inference with the MCMC sampling algorithm for parameter estimation. Andrieu et al. [5] have an interesting discussion on this promising frontier.

2.4 HMRF Model

2.4.1 the HMRF Model

The Hidden Markov Random Field (HMRF) model is a special case of an undirected graphical model. Here we consider an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges of graph G. Let $X = \{x_i, i \in V\}$ be the set of latent random variables associated with nodes V. For each node i, x_i is linked with visible random variable y_i ($y_i \in \mathbf{R}$), where $y_i = \hat{y}_i$ refers to current observed value. Let $E = \{\mathcal{E}, \hat{\mathcal{E}}\}$ with $\hat{\mathcal{E}}$ the set of edges between X and Y, $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ the set of edges within X.[34] To simplify the notation, we use $|V|$ for the number of edges, $|\mathcal{E}|$ for the number of edges within X, upper case letters $\{X, Y\}$ for the set of node configurations over G, lower case letters $\{x_i, y_i\}$ or $\{x, y; x = x_i, y = y_i, \forall i \in V\}$ for single node configurations, $\{x_e, y_e\}$ for the set of nodes configuration over an edge of G, and (i, j) for $e(i, j)$. $\theta = (\alpha, \beta)$ denotes the parameter set with α associated with $\mathcal{E}$, and β with $\hat{\mathcal{E}}$.

The Hammersley-Clifford theorem B.6, enables that the probability functions to be within the exponential family, with the potential functions $\phi_e(x_e)$ and $\phi_i(x_i, y_i)$ corresponding to the natural parameters α and β, respectively. From the structure of G, assume X is a Markov Random Field (MRF, Appendix B.6) while Y is a random field. We can now define the HMRF model by enforcing the constraints as follows:

- In general, the domain of x could be either discrete $x \in \{1, \ldots, K\}$, or continuous $x \in \mathbf{R}$. The domain of X is $x^{|V|}$.[35]
- The parameter vector α is tied [36] over all edges $\alpha = \alpha_e, \forall e \in \mathcal{E}$. The set of parameter vectors $\beta = \{\beta_x\}$ is also tied $\beta = \beta_i, \forall$ edge $(x_i, y_i) \in \hat{\mathcal{E}}$, and is

[34] Usually in the HMRF model, the X is arranged as a two-dimensional lattice. The edge $e(i, j) \in \mathcal{E}$ connects a pair of nodes in the nearest neighbour fashion, over the MRF X. The edge $e(i, i) \in \hat{\mathcal{E}}$ links the hidden node x_i and the corresponding observation $\hat{y}_i$.

[35] To consider both cases, we resort to the σ-finite measure ν on x (X).

[36] This tying of parameters over nodes is called **standard** HMRF, which is in contrast with the **generalized** HMRF where parameters (α or β) are different over nodes. Here we consider only the **standard** HMRF case, and call it HMRF throughout.

x-dependant

- when x is discrete, $\beta_x = \beta_k$, where $\beta_k \in \{\beta_{x1}, \ldots, \beta_K\}$ is a parameter vector,
- when x is continuous, $\beta_x = \beta(x)$ is a parameter vector as a function of x.

- The potential functions are tied in the same way: $\phi(x_e) = \phi_e(x_e), \forall e \in \mathcal{E}$; $\phi(x_i, y_i) = \phi_i(x_i, y_i), \forall i \in V$.
- The conditional function $y|x$ is assumed as *iid* distribution.

Given the above settings, we can define the probability functions. Define the empirical conditional probability function over a single node x as:

$$p(\hat{y}|x, \beta_x) = \exp\{< \beta_x, \phi(x, \hat{y}) > -Z(\beta_x)\}$$

where $< \cdot >$ denote the inner product. The log partition function is

$$Z(\beta_x) = \log \int_y \exp\{< \beta_x, \phi(x, y) >\} dy.$$

Thus the empirical conditional probability function over domain X is

$$p(\hat{Y}|X, \beta) = \prod_i p(\hat{y}_i|x_i, \beta_{x_i}) \tag{2.66}$$

$$= \exp\{\sum_{i \in V} (< \beta_{x_i}, \phi(x_i, \hat{y}_i) > -Z(\beta_{x_i}))\}. \tag{2.67}$$

Notice that

$$\sum_i \log p(\hat{y}_i|x_i, \beta_{x_i}) = \sum_{i \in V} (< \beta_{x_i}, \phi(x_i, \hat{y}_i) > -Z(\beta_{x_i})).$$

Also, define the probability function for X

$$p(X|\alpha) = \exp\{< \alpha, \sum_e \phi(x_e) > -Z(\alpha)\} \tag{2.68}$$

where the log partition function $Z(\alpha) = \log \int_X \exp\{< \alpha, \sum_e \phi(x_e) >\} \nu(dX)$. Therefore, we can set up the joint probability as:

$$p(X, \hat{Y}|\theta) = p(X|\alpha) p(\hat{Y}|X, \beta) \tag{2.69}$$

As a concrete example, consider the Potts model employed for modelling MRF X, [37] that is, any random variable x_i associated with node $i \in V$ obeys the multinomial

[37] The Potts model is a generalization of the Ising model which $x_i \in \{0, 1\}; i \in V$.

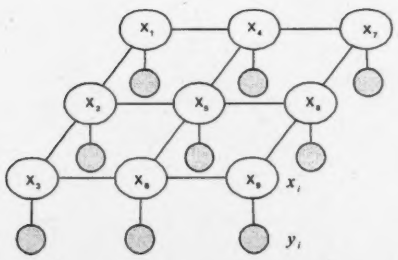


Figure 2.4: An exemplar 3×3 HMRF model. The white nodes are the latent variables x, and the grey nodes are the observed variables y. Associated with any edge e connecting two neighbouring latent nodes (x_i, x_j), there is the function represented as $\phi(x_e)$. Similarly, for any (x_i, y_i) pair there is an associated function represented as $\phi(x_i, y_i)$.

distribution, $x_i \in \{1, 2, \ldots, K\}$. Thus $\sum_e \phi(x_e) = \sum_{(i,j)\in E} x_i x_j$ where $x_i x_j = 1$ if $x_i = x_j$ and 0 otherwise. Then assume $Y|X$ obeys the Gaussian pdf, [38] with each Gaussian pdf for $y|x$ given the parameter β_x. In its exponential form, the natural parameter for Gauss distribution is $\beta_x = (\beta_{x1}, \beta_{x2})$, the single node conditional probability function

$$p(y|x, \beta_x) = \exp\left\{\beta_{x1} y - \frac{1}{2}\beta_{x2} y^2 - Z(\beta_x)\right\},$$

with the log partition function

$$Z(\beta_x) = \frac{1}{2}\left(\frac{\beta_{x1}^2}{\beta_{x2}} + \log\frac{2\pi}{\beta_{x2}}\right). \tag{2.70}$$

By the *iid* assumption of the HMRF model, the conditional function is factorized

$$p(\hat{Y}|X, \beta) = \exp\left\{\sum_X \left(\sum_i \delta(x_i, x) < \beta_{x1}, \hat{y}_i > -\frac{1}{2}\sum_i \delta(x_i, x) < \beta_{x2}, \hat{y}_i^2 > -|V_x| Z(\beta_x)\right)\right\}$$

and the probability function for Potts model X is

$$p(X|\alpha) = \exp\{\alpha \sum_{(i,j)\in E} x_i x_j - Z(\alpha)\}. \tag{2.71}$$

The joint probability function is computed according to Eq. (2.69).

Later we will come back to this example and derive the specific updating equations.

[38] Any distribution could be used for $y|x$ as long as it falls within the exponential family. For example, it is straightforward to generalize the $Y|X$ distribution to a multivariate Gaussian pdf.

2.4.2 Relationships with the Boltzmann Machine and the Hidden Markov Model

The HMRF model clearly belongs to the family of the Undirected Graphical Models (UGM) [48]. In particular, the HMRF can be regarded as a special case of the Boltzmann Machine (BM) [2]. Given a UGM over variables s, the Boltzmann Machine can be defined as

$$p(s|W) = \exp\left\{\sum_{i \neq j} W_{i,j} s_i s_j - Z\right\}$$

where W is a symmetric matrix with zero-diagonal elements, and Z is the log-partition function. For the HMRF model described in Sec. 2.4.1, define the complete data $s = (x, y)$ and write the tied parameters (α, β) in matrix form W, as (here we show the W matrix for Fig.2.4)

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ y_1 \\ \vdots \\ y_9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & y_1 & y_2 & y_3 & \cdots & y_9 \end{array} \\
\left(\begin{array}{ccccccccccccc}
 & \alpha & & \alpha & & & & & & \beta_{x1} & & & & \\
 & & \alpha & & \alpha & & & & & & \beta_{x2} & & & \\
 & & & & & \alpha & & & & & & \beta_{x3} & & \\
 & & & & \alpha & & \alpha & & & & & & & \\
 & & & & & \alpha & & \alpha & & & & & & \\
 & & & & & & & & \alpha & & & & & \\
 & & & & & & & \alpha & & & & & & \\
 & & & & & & & & \alpha & & & & & \\
 & & & & & & & & & & & & & \beta_{x9} \\
 & & & & & & & & & 0 & & \cdots & & 0 \\
 & & & & & & & & & & & \ddots & & \vdots \\
 & & & & & & & & & & & & & 0
\end{array}\right)
\end{array}$$

where the α value is tied over all latent nodes x, β values are node x dependent. For simplicity, only the upper diagonal half is considered, since W is symmetric. Also, the non-zero entries are specified with the values while empty entries refer to zero-valued entry.

Obviously as shown in this example, the HMRF model could thus be modelled as a Boltzmann Machine with latent and continuous variables, with sparse and highly structured W matrix. Notice that the β parameter is tied over nodes within the same type, and is dependent on the estimation of x values. Besides, this $2|V| \times 2|V|$ W

matrix can be conveniently decomposed into three square sub-matrices: the $|V| \times |V|$ $x-x$ banded matrix, the $|V| \times |V|$ $x-y$ diagonal matrix, and the $|V| \times |V|$ $y-y$ zero matrix. This connection points out the potential for solutions to the HMRF model by taking advantage of the existing methods for solving the Boltzmann Machine, and combining fast numerical algorithms for solving sparse and highly structured matrices.

From the Expectation–Maximization (EM) perspective, the HMRF could be viewed as 2D generalization of the Hidden Markov Model (HMM) [72] with the major difference being that the HMRF is defined over a 2D lattice rather than a 1D sequence for HMM. This difference, however, leads to the intractability [39] of finding the exact solution in the E-step in the HMRF, while in general it is easy to find an exact solution in the E step in HMM. Therefore, in HMRF, the E step is an approximation step, and deterministic algorithms like the LBP algorithm could be employed to find the max-product solution for determining x, which corresponds to the belief propagation (BP) [48] algorithm, [40] (or equivalently, Viterbi algorithm, refer to Jordan et al. [48]) which is employed to infer the max-product solution of x in HMM. Also, the two models may also be slightly different in the M step. In HMRF, we often end up with the fixed point solutions (iterative equations) for parameter estimation. In HMM, we can derive the closed form solutions instead. From this connection, researchers have tried to approximate HMRF by augmenting HMMs to 2D, as for example, the approach by Li et al. [47].

2.4.3 Algorithms for Parameter Estimation

The variational EM algorithm

Formulated as a variational problem, the EM algorithm is interpreted as an alternative maximizing of the variational lower bound $\mathcal{L}(\mu(q), \theta)$[41]:

E step

$$\mu^{(t+1)}(q) = \arg\max_{\mu(q)} \mathcal{L}(\mu(q), \theta^{(t)}) \tag{2.72}$$

$$= \arg\max_{\mu(q)} \{< \mu(q), \theta > -Z^*(\mu(q)) - Z_q(\beta)\} \tag{2.73}$$

[39] Recall that the loopy structure of this UGM renders finding the exact solution an NP hard problem.

[40] The Loopy Belief Propagation (LBP) algorithm is the loopy version of BP. We will discuss LBP in detail in Sec. 4.2.1.

[41] We define the function of mapping $q \to \mu$ as $\mu = \mu(q)$.

M step

$$\theta^{(t+1)} = \arg\max_{\theta} \mathcal{L}(\mu(q)^{(t+1)}, \theta) \tag{2.74}$$

$$= \arg\max_{\theta} \{< \theta, \mu(q)^{(t+1)} > - Z(\alpha) - Z_q(\beta)\} \tag{2.75}$$

Since both lower bounds are concave functions, the critical points can be computed iteratively by computing the derivatives and equating to zero, respectively. [42] In the E step, the update rule is of closed form

$$q(X) = \exp\{< \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i}) - Z(\theta)\}. \tag{2.76}$$

In the M step, the closed form solution for α does not exist. In general, α can be updated either by the gradient descent or by the iterative scaling method. In the gradient descent method the search direction is:

$$\frac{\delta \mathcal{L}^{\mathcal{M}}(\alpha)}{\delta \alpha} = E_{p(X|\alpha)}[\sum_e \phi(x_e)] - E_q(X)[\sum_e \phi(x_e)]. \tag{2.77}$$

In our example, $\beta_x = (\beta_{x1}, \beta_{x2})$ is updated by finding the roots of the equations

$$E_q\left[\sum_i (\hat{y}_i|_{x_i=x})\right] = E_q\left[|V_x|\frac{\beta_{x1}}{\beta_{x2}}\right] \tag{2.78}$$

$$E_q\left[\sum_i (\hat{y}_i^2|_{x_i=x})\right] = E_q\left[|V_x|\frac{\beta_{x1}^2}{\beta_{x2}^2} + \frac{1}{\beta_{x2}}\right] \tag{2.79}$$

Variational approximate in the E step

Up to now, we executed an exact inference algorithm in the E step. In practice, the computation complexity for evaluating the function $q(X)$ grows exponentially with $|V|$, the number of nodes in graph G, for a HMRF model with reasonable size. Therefore, we could only resort to approximation methods which include, for example, the family of mean field algorithms. [43]. The essence of the mean field algorithms is, instead of computing the exact probability function $q(X)$, a factorability assumption is imposed to approximating $q(X)$ by $q_{\mathcal{H}}(X) = \prod_{h \in \mathcal{H}} q_h(x_h)$, resulting in the variational E step:

$$\mu^{(t+1)}(q_{\mathcal{H}}) = \arg\max_{\mu(q_{\mathcal{H}})} \mathcal{L}(\mu(q_{\mathcal{H}}), \theta^{(t)}) \tag{2.80}$$

$$= \arg\max_{\mu(q_{\mathcal{H}})} \{< \mu(q_{\mathcal{H}}), \theta > - Z^*(\mu(q_{\mathcal{H}}))\}. \tag{2.81}$$

[42] Detailed derivations are referred to Appendix B.8.

[43] There exist other approximation algorithms, such as loopy belief propagation and its variants [91] In this section we only focus on the mean field algorithms.

It can be proved [90] that the family of mean field approximations will approximate the lower bound $\mathcal{L}(\mu(q), \theta^{(t)})$ in the E step by $\mathcal{L}(\mu(q_{\mathcal{H}}), \theta^{(t)})$, with

$$\mathcal{L}(\mu(q_{\mathcal{H}}), \theta^{(t)}) \leq L(\theta), \tag{2.82}$$

with equality holding when $E_{\mathcal{H}} = E$ and

$$q_h(X_h) = p(X_h|Y, \theta) \quad \forall h \in \mathcal{H}. \tag{2.83}$$

2.4.4 The Mean Field Algorithms

To perform an approximate inference in the E step, we consider the family of mean field algorithms. Given a graph G, we define a subgraph $G_{\mathcal{H}} = (V, E_{\mathcal{H}})$, with $E_{\mathcal{H}} \subseteq E$. That is, by deliberately removing some edges from the original graph G, [44] we can always form a subgraph $G_{\mathcal{H}}$ which may be more computationally tractable, with the cost of loosing some structure from G.

Naive Mean Field algorithm

Consider the simplest case when all edges are removed from G, forming a subgraph $G_{\mathcal{H}_0} = (V, E_{\mathcal{H}_0})$ with $E_{\mathcal{H}_0} = \varnothing$. This is the Naive Mean Field (NMF) setting, with the probability function

$$q(X) = \prod_{i \in V} q(x_i). \tag{2.84}$$

In this NMF approximation, since $E_{\mathcal{H}_0} = \varnothing$, for any node i and its neighbouring node j[45], the correlation term $x_i x_j$ is replaced by $x_i \mu_j$ where μ_j is the estimated mean value of x_j. Thus, in our example, Eq. (2.71) becomes[46]

$$q(x_i) = \exp\left\{\alpha\left(\sum_{j \in \partial i, j > i} \mu_j\right) x_i + \beta_{x_i 1}\hat{y}_i - \frac{1}{2}\beta_{x_i 2}\hat{y}_i^2 - Z_i(\theta)\right\}. \tag{2.85}$$

$\mathcal{L}^M_{\mathcal{H}_0}(\alpha)$ can be derived according to Eq. (B.23) (see Appendix B.8). Compute α as the gradient descent of

$$\frac{\delta \mathcal{L}^M_{\mathcal{H}_0}(\alpha)}{\delta \alpha} = \sum_i E_{p(x_i|\alpha)}[x_i \mu_{j, j \in \partial i}] - \sum_i E_{q_{\mathcal{H}_0}(x_i)}[x_i \mu_{j, j \in \partial i}]. \tag{2.86}$$

[44] Here we only consider the removal of the edges within X, that is, let $E_{\mathcal{H}} = \{\mathcal{E}, \hat{\mathcal{E}}\}$, the edges $e(i, i) \in \hat{\mathcal{E}}$ are preserved by default.

[45] j is the neighbouring node of i, when $(i, j) \in E$. Sometimes we also use the notation $j \in \partial i$.

[46] Refers to Appendix B.9 for details.

The NMF algorithm could be deemed as one deterministic variant of the Gibbs sampler. Another deterministic variant of the Gibbs sampler is the Iterative Conditional Mode (ICM). When conditioned on the neighboring nodes and we want to capture the posterior distribution of the current node, i, the Gibbs sampler draws an amount of samples to capture this distribution. NMF approximates the node-wise interaction by a linear function $x_i x_j \approx x_i \mu_j$, then approximates the factorized distribution by optimizing the lower bound of the KL divergence. The ICM computes the MAP estimation, based on the same condition as the Gibbs sampling.

In the same manner, $\mathcal{L}^M_{\mathcal{H}_0}(\beta)$ can be formulated from Eq. (B.24) (see Appendix B.8). This leads to the closed form solution for β_x as

$$\beta_{x1} = \frac{1}{E_q[\hat{Y}^2/\hat{Y}] - (E_q[\hat{Y}])} \tag{2.87}$$

$$\beta_{x2} = \frac{1}{E_q[\hat{Y}^2] - (E_q[\hat{Y}])^2} \tag{2.88}$$

with

$$E_q[\hat{Y}] = \frac{\sum_i q(x_i)\hat{y}_i}{\sum_i q(x_i)}, \tag{2.89}$$

$$E_q[\hat{Y}^2] = \frac{\sum_i q(x_i)\hat{y}_i^2}{\sum_i q(x_i)} \tag{2.90}$$

$$E_q[E_q[\hat{Y}^2/\hat{Y}] = \frac{\sum_i q(x_i)\hat{y}_i^2}{\sum_i q(x_i)\hat{y}_i}. \tag{2.91}$$

Therefore, the NMF–EM procedure is

- (E step) Inference
  - Infer the configuration over X by computing Eq. (2.85) over every node i, iteratively.
- (E step) Sufficient statistics
  - Compute the sufficient statistics of Eqs. (2.89), (2.90), (2.91), and that of $\sum_i E_{q_{\mathcal{H}_0}(x_i)}[x_i \mu_j]$ which appeared in the LHS of Eq. (2.86).
- M step
  - Update parameters θ, by Eq. (2.86), Eq. (2.87)and Eq. (2.88).

In our example, the detailed NMF–EM procedure is

- E step

- the lower bound

$$\mathcal{L}^E(\mu) = E_q[\alpha x_i x_j] + E_q[< \beta, \phi(x_i, \hat{y}_i) >] - Z^*(\mu)$$

- the update equation

$$q(x_i) = \exp\{\alpha x_i \sum_{j \in \partial i} \mu_i + \beta_{x1}\hat{y}_i - \frac{1}{2}\beta_{x2}\hat{y}_i^2 - Z_i(\theta)\}.$$

- M step

  - the lower bound

$$\mathcal{L}^M(\theta) = E_q[\alpha x_i x_j] + E_q[< \beta, \phi(x_i, \hat{y}_i) >] - Z(\alpha)$$

  - update α as root-finding of the equation

$$E_p[x_i x_j] = E_q[x_i x_j]$$

  - update β in the closed form solutions, by Eq. (2.87) and Eq. (2.88).

Where

$$E_q[\alpha x_i x_j] = \sum_{(i,j) \in E} \sum_{x_i, x_j} q(x_i) q(x_j) \alpha x_i x_j$$

$$E_q[< \beta, \phi(x_i, \hat{y}_i) >] = \sum_i \sum_{x_i} q(x_i)(\beta_{x1}\hat{y}_i - \frac{1}{2}\beta_{x2}\hat{y}_i^2 - Z(\beta, x_i))$$

$$Z_i(\theta) = \sum_{x_i} \exp\{\alpha x_i \sum_{j \in \partial i} \mu_i + \beta_{x1}\hat{y}_i - \frac{1}{2}\beta_{x2}\hat{y}_i^2\}$$

$$Z^*(\mu) = \sum_i \sum_{x_i} q(x_i) \log q(x_i)$$

$$Z(\alpha) = \sum_{(i,j) \in E} \log \sum_{x_i, x_j} \exp\{\alpha x_i x_j\}$$

$$E_p[x_i x_j] = \sum_i \sum_{j \in \partial i} p(x_i|\alpha) x_i \mu_j$$

$$E_q[x_i x_j] = \sum_i \sum_{j \in \partial i} q(x_i) x_i \mu_j.$$

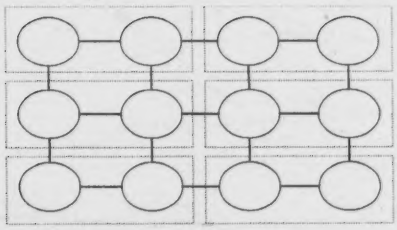


Figure 2.5: An example of the GMF algorithm. Here we show a 3×4 2D graph of the HMRF. By applying the GMF algorithm, the graph is divided into 6 subgraphs, with each subgraph of the size 2×1, which can be regarded as a meta-node. See text for details.

Generalized Mean Field (GMF) Algorithm

The idea of NMF could be further generalized to preserve certain tractable subgraph structure during the factorization. This result in Generalized Mean Field (GMF) algorithm [90] that could be regarded as a deterministic version of the block-Gibbs sampler. As illustrated in Fig.2.5, each subgraph containing a number of spatially clustered nodes is treated as a meta-node. For nodes within the meta-node, exact inference is applied. For the meta-level MRF, the NMF algorithm is applied. Notice that when the meta-nodes boil down to contain a single node, the GMF will be equivalent to the NMF.

2.5 Summary

In this chapter we have introduced some key concepts on related statistical models and methods, which are the building blocks of the proposed FORIUS system that follows.

Chapter 3

The Segmentation and Labelling Component

The first important step of any image interpretation system is feature or region extraction and labelling. *The aim is to derive the optimal segmentation and labelling of pixels given a small set of annotated observations.* In another word, image segmentation and labelling is formulated as a single, trainable, integrated process. In this chapter, after surveying the related literature,[1] we describe our proposed supervised segmentation method, make comparisons with one state-of-art method (Bouman et al. [20]) on the document image data that they adopted, and report some experiments on our forestry image data.

3.1 The Image Model: Hierarchical Markovian Models

In recent years there have been significant developments in formulating image encoding, segmentation, and labelling using hierarchical Markov models. With such models image annotation is viewed as defining labelling operators over sets of hierarchical (multiscaled) Markov Random Fields (MRF) with the use of techniques from Hidden Markov Models (HMM) such as dynamic programming and Expectation Maximization (EM) [33, 83, 12, 58, 54, 13, 20]. These models, as described below, include Multi-Scale Markov Random Field (MSRF) models [13, 20], Tree-Structured Bayesian Networks [89, 3], Hidden Markov Trees (HMT) [23, 22], and 2D Multi-resolution HMMs [47] etc.

The MSRF model for image labelling by Bouman and his colleagues [13] utilizes quadtree-structured labelling layers X^0, X^1, ... , X^L, associated with the observed image, and a sequential intra-layer maximum posterior probability criterion (Sequential MAP) for inference at each layer. The Expectation–Maximization (EM) algorithm was also applied to estimate the model parameters $\Theta_n = (\Theta_{n,0}, \Theta_{n,1}), n \in (0..L-1)$, where Θ_n defined the balance between deterministic averaging and stochastic sampling analogous to the fudge factor λ in the Levenberg–Marquardt method [71]. To address the limitation of such unsupervised procedures, Cheng et al. [20] extended it to supervised learning where, in addition to the quadtree labelling layers, they also used corresponding Haar wavelet coefficients layers. Finally, a regression tree was utilized to capture the intra-layer class-transition probabilities. Satisfactory results were obtained by applying this model to document grey-level

[1] There is a vast amount of literature on image segmentation with methods developed from different disciplines (mathematics, statistics, physics). Here our attempt is highly biased towards reviewing the statistical Markovian methods which are most relevant to our proposed algorithm.

image segmentation applications.

The wavelet-based HMT model proposed by Crouse, Nowak and Baraniuk [23] is an extension to Hidden Markov Models (HMM) by framing the quadtree hierarchy as a HMM, where each state can have several concurrent child states. In this case the "tying" technique [72] is extensively utilized to reduce the amount of unknown parameters. Choi et al. [22] applied the HMT to image segmentation where the HMT was used to model the intra-layer dependency of the quadtree-structured wavelet coefficients. Further, a "contextual labelling tree" was introduced as an enhancement of the HMT to avoid the "blocky" artifact effects, a common drawback with the quadtree structure. Simulations on remotely sensed data and document images were performed using this generative model.

The 2D HMM of Gray and Li [47] extended 1D HMMs directly, and used numerous constraints to make the parameter estimation and inference from the 2D HMM engine tractable. As an extension, a multi-resolution 2D HMM [47] was developed in which the intra-layer dependency was addressed either jointly (optimal solution guaranteed while slower) with, or separately (sub-optimal solution but faster) from, the inter-layer dependencies.

The above-mentioned models have advantages under certain conditions. However, they all only deal with grey-scale images in contrast to the multi-spectral forestry inventory data we are faced with here. Also, because all of them apply to wavelet domains using quadtree representations, "blocky" artifact effects are unavoidable. Solutions have been proposed to alleviate this problem and include "contextual labelling tree" for the HMT model and the "pyramidal graph model" for the MSRFs. These methods (HMT and MSRF) utilize upward or downward constraint propagation methods — but not both — as we examine here in our proposed bijective operators. Further, in none of these approaches are the parameters tuned to best fit expert annotated data.

So, in general, what differentiates our approach [2] (the basic model shown in Fig. 3.1) from the aforementioned is that we explore

1. a flexible probabilistic framework for the segmentation of either grey or multi-spectral images, using either Gaussian or Wavelet pyramid models,

2. a form of hierarchical constraint propagation using bijective operators,

[2] A preliminary version of this segmenter-annotator appeared in [21].

3. the tuning of the model parameters to best fit expert annotation.

3.2 The Hierarchical Hidden Markov Tree Model

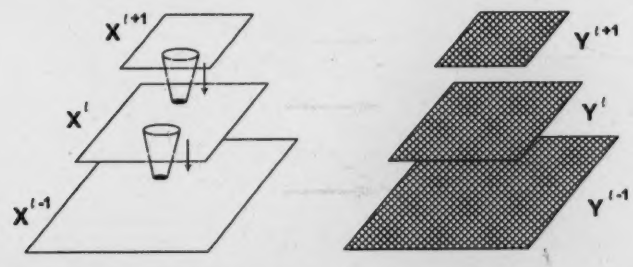


Figure 3.1: The basic Hierarchical Hidden Markov Tree (HHMT) model. Here only three levels ($l-1$, l,$l+1$) of the multiscale representation are shown. At each level there are two random fields corresponding to observed pixels, Y, image region labels defined by X. The bijection operations shown in grey and black colors (with corresponding arrows) represent upward and downward support kernel operators and associated region label transitions over movements upwards and downwards in scale, encapsulating two types of contextual constraints.

Following Bouman et al. [12], the observed image Y is defined as a MRF, and it is seen as an instance from an underlying stochastic process, X, which is also a MRF. This HMRF model has been used in medical image applications [93], but it has the disadvantage of not explicitly encoding long-range correlations between pixels. So this basic model has been extended to Hierarchical Hidden Markov Tree (HHMT) model which encode such multi-scaled information using image pyramid data models and, as we will see, a bijective operator which can propagate constraints over larger areas of the image. In this model (see Fig. 3.1), each layer l consists of two random fields to encode the hidden labelling process (X^l) and the observed image (Y^l). The ith site (pixel) for the hidden and observation random fields are denoted by x_i^l and y_i^l, respectively.

Since the posterior (label) probability at any given layer l, $p(X^l|Y^l)$ [3] is computationally intractable, we introduce further assumptions which enable us to perform parameter estimation in a feasible way. These are:

1. In any given layer l, the observed random field Y^l is solely dependent on the hidden states at the same level, X^l:

$$p(Y^l|X^{ALL}, Y^{ALL}) = p(Y^l|X^l) \tag{3.1}$$

[3] We put the nomenclature of this chapter in Appendix A.1.

where X^{ALL}, Y^{ALL} refer to the complete labelling/observation hierarchy. Further, each observed data pixel is dependent only on its corresponding state (label)

$$p(Y^l|X^l) = \prod_i p(y_i^l|x_i^l) \tag{3.2}$$

2. The intra-layer hidden states are only dependent on their adjacent layers, such as:

$$p(X^l|X^{ALL}) = p(X^l|X^{\partial l}) \tag{3.3}$$

where X^{ALL} refers to the whole hidden hierarchy, and $X^{\partial l}$ corresponds to the neighborhood layers of X^l, i.e., X^{l-1}, and X^{l+1}.

3. In each layer l, the inter-layer hidden states (labels) are independent of each other, that is:

$$p(X^l|Y^l) = \prod_i p(x_i^l|y_i^l). \tag{3.4}$$

Consequently, the proposed HHMT model (Fig. 3.1) is defined as

$$\lambda = \{(\pi_l, A^+_{l-1,l}, A^-_{l,l-1}, B_l); l = 1, .., L\} \tag{3.5}$$

where the upward state transition matrices A^+ are defined by

$$A^+_{l-1,l}(\vec{i}, j) = p(x_s^{(l)} = j/\hat{\partial s}^{(l-1)} = \vec{i}). \tag{3.6}$$

The downward transition matrices A^- are defined by

$$A^-_{l,l-1}(\vec{i}, j) = p(x_s^{(l-1)} = j/\hat{\partial s}^{(l)} = \vec{i}). \tag{3.7}$$

For position j in layer l, $\hat{\partial s}^{(l+1)} = \vec{i}$ defines its contextual parents (clique, kernel), and $\hat{\partial s}^{(l-1)} = \vec{j}$ corresponds to its contextual children. In both cases, these kernels are defined by indexed histograms (see below).

For each layer the prior probabilities of each label are defined by

$$\pi_i^l = p(X^l = i), \tag{3.8}$$

corresponding to the relative frequencies of expected image labels at a given scale, l. The observation matrices B^l are defined by $B \equiv \{B^l, l \in [1..L]\}$, where $B^l(o, c) \in B^l$ characterize the likelihood of the observed pixel values (o) at level l, given label

$c \in [1..C]$. Such conditional probabilities are derived from a mixture-based clustering model in the label-dependent colour cluster space and corresponds, after normalization, to

$$B^l(o,c) = p(y^l = o/x^l = c). \tag{3.9}$$

Specifically, the clustering method used here follows Bouman's Minimum Description Length (MDL [74]) criterion based mixture of Gaussian method [13] for choosing the number of clusters. The objective is to minimize the MDL criterion given by

$$MDL(K,\theta) = -\sum_{n=1}^{N} \log\left(\sum_{i=1}^{K} p_{y_n|x_n}(y_n|k,\theta)\pi_k\right) + \frac{1}{2}S\log(NM) \tag{3.10}$$

where K is the number of Gaussians, and

$$\theta = \{(\{\pi_k\}_{k=1}^K, \{\mu_k\}_{k=1}^K, \{C_k\}_{k=1}^K)\}$$

corresponds to the priors, means, covariance matrices for the K Gaussian mixture, respectively, and

$$S = K\left(1 + M + \frac{(M+1)M}{2}\right) - 1.$$

From a Bayesian point of view, Eq. (3.10) is composed of two parts, the likelihood part and the prior part. For simplicity of computation we maximize the function

$$Q(K,\theta) \propto MDL(K,\theta)$$

where:

$$\begin{aligned} Q(K,\theta) = & \\ & \sum_{k=1}^{K} \bar{N}_k(-\frac{1}{2}trace[\bar{R}_k R_k^{-1}] - \frac{1}{2}(\bar{\mu}_k - \mu_k)^t R_k^{-1}(\bar{\mu}_k - \mu_k) - \frac{M}{2}\log(2\pi) \\ & -\frac{1}{2}\log(|R_k|) + \log(\pi_k)) - \frac{1}{2}L\log(NM) \end{aligned}$$

for $\bar{N}_k$, $\bar{\mu}_k$, and $\bar{R}_k$ being

$$\begin{aligned} \bar{N}_k &= \sum_{n=1}^{N} p_{x_n|y_n}(k|y_n,K,\theta) \\ \bar{\pi}_k &= \frac{\bar{N}_k}{N} \\ \bar{\mu}_k &= \frac{1}{\bar{N}_k}\sum_{n=1}^{N} y_n p_{x_n|y_n}(k|y_n,K,\theta) \\ \bar{R}_k &= \frac{1}{\bar{N}_k}\sum_{n=1}^{N} (y_n - \bar{\mu}_k)(y_n - \bar{\mu}_k)^t p_{x_n|y_n}(k|y_n,K,\theta) \end{aligned}$$

using the Expectation-Maximization (EM) algorithm (see Bouman et al. [13] for more details). It should be noted that we have used a stochastic version of the Expectation Maximization (SEM) procedure (see Celux et al. [19]), to evaluate the Gaussian mixture [60] parameters and, for the coloured images used here, a multivariate Gaussian mixture model.

Optimization problem I: Initial pixel labelling: Select pixel labels that maximize the following MAP criterion:

$$X^l = \arg\max_{X^l,k}(p(Y^l|X^l, G_k)p(X^l)). \tag{3.11}$$

Optimization problem II: Segmentation and Region Labelling. Given a model estimation procedure, initial estimates of pixel labels, image segmentation and labelling reduces to that of determining the most likely pixel labelling over scales, given the model and observed images:

$$X^l = \arg\max_{X^l,k}(p(Y^l|X^l, G_k)p(X^l|X^{\partial l})p(X^l)). \tag{3.12}$$

3.2.1 The Estimation Algorithm

The initial estimates of the model are obtained from expert annotated training images using the following procedure.

1. For each training image, construct a Gaussian pyramid (where layer 1 to layer L correspond to the finest to coarsest levels, respectively).

2. The finest layer is manually labelled, and then sub-sampling (consistent with the upper frequency of each pyramid layer) is performed to obtain initial labels for the upper layers.

3. The prior initial estimates of label probabilities at a given scale as determined from these statistics.

4. In the downward scanning phase, the transition matrix $A^-_{l+1,l}$ is determined by moving the kernels over the complete image layers $l+1$ and l, and continues in this direction until the finest layer is reached. A similar method is used in the upward scanning phase to determine $A^+_{l-1,l}$.

5. The state-dependent observation matrices are determined from the Gaussian mixture models for each class c and layer l, using the MDL mixture of Gaussians clustering model.

Once this initial model is obtained from the training images, the process of image segmentation and labelling involves instantiating (and updating) the model from new image data as follows.

3.2.2 The Segmentation and Labelling Algorithm

To interpret (annotate) new images, upward-downward recursion is used to propagate scene labels across layers until they are as compatible as possible with respect to the support kernel statistics. This is similar in spirit to the forward-backward loop in traditional HMMs and from this model the general annotation rules are derived from:

$$X^l = \arg\max_{x \in X} p(X^l / Y^l). \tag{3.13}$$

Due to the assumption of independence we have, for image I,

$$X^l = \Pi_{i \in I} x_i^l \tag{3.14}$$

for

$$x_i^l = \arg\max_{c \in X^l} p(x_i^l = c / y_i^l) \tag{3.15}$$

where

$$p(x_i^l = c / y_i^l) = \lambda p(y_i^l / x_i^l = c) p(x_i^l = c / \hat{\partial s}^{(l)}). \tag{3.16}$$

Here, λ is a normalizing factor and

$$p(y_i^l / x_i^l = c) = \max_{m \in M} \alpha_m g(y_i^l; \mu_{c,m}^l, \sigma_{c,m}^l) \tag{3.17}$$

where α_m is the estimated weight for cluster m in the label-dependent observation space, over M Gaussian mixtures, $g(*)$. This formulation uses a MAP criterion for determining the cluster membership for a given observation (pixel value) and the underlying Gaussian mixture model (see Section 3.2).

Given these relations, the actual upward-downward procedure is as follows:

1. For a candidate image, generate the Gaussian pyramid.
2. Naive Bayesian classifier: Use the MAP principle, based purely on the B matrix and π, to obtain an initial labelling of layer L (the coarsest layer):

$$\hat{x_i^L} = \arg\max_{x_i^L \in X^L} p(x_i^L / y_i^L)$$

$$p(x_i^L/y_i^L) = p(y_i^L/x_i^L) * p(x_i^L).$$

3. Downward recursion using π, A^- and B from the coarsest layer L to the finest layer 1

$$\hat{x_i^l} = \arg\max_x p(x_i^l/y_i^l)$$
$$p(x_i^l/y_i^l) = p(y_i^l/x_i^l)p(x_i^l/\hat{\partial s_i}^{(l+1)}).$$

4. Upward recursion using π, A^+ and B, from the finest layer 1 to the coarsest layer L

$$\hat{x_i^l} = \arg\max_{x \in X} p(x_i^l/y_i^l)$$
$$p(x_i^l/y_i^l) = p(y_i^l/x_i^l) * p(x_i^l/\hat{\partial s_i}^{(l-1)}).$$

5. Iterate from step 3 to 4, and compute the log-likelihood score:

$$Q^l = \sum_i \log \sum_c p(x_i^l = c/y_i^l, \hat{\partial s_i}) \tag{3.18}$$

$$Q^* = \sum_{l=1}^{L} \gamma^l Q^l$$

where

$$\gamma^l = D^{2\times(l-1)}/P.$$

D is the size of the kernel(clique), P is the number of pixels, Q^l is the log-likelihood score for current layer l, and Q^* is for the image over all layers.

6. Repeat 1–5 until no label changes over all layers.

The inter-layer contextual (upper and lower) dependencies are computed from the conditional histograms corresponding to a given label over each particular pair of image scales such that each code in the histogram code-book is formed as a context vector. During training, this is computed from the complete set of training images for both upward and downward codebooks, (see Fig. 3.2) for each consecutive pair of levels where we extract the $p(x/\partial s)$ quantities. The codebook [4] serves to index

[4] In theory, the size of the code-book will grow exponentially with the number of classes. We plan to approximate the codebook with regression model, to address the scaling-up problem.

		l	
		1	2
l+1	18	0.1	0.9
	45	0.6	0.4
	54	0.3	0.7
	81	0.9	0.1

(a)

		l	
		1	2
l-1	18	0.2	0.8
	45	0.6	0.4
	54	0.3	0.7
	81	0.1	0.9

(b)

Figure 3.2: A simple example of the indexed histogram codebook, with a window of size 3×3. Assume we have two class of labels, e.g. $C = \{1, 2\}$, at all levels, and we are in layer l. The numbers in the two most left columns in (a) and (b) represent the frequencies of class 1 and 2 for level $l + 1$ (a) over the image while the probabilities correspond to the relative (conditional) frequencies of occurrence of labels 1 and 2 at the next level l (a). (a) is the table for downward operation, and (b) is the table for the upward operation.

all labels between neighbouring layers (up and down) for a fixed kernel size over all scales of the annotated image. In the labelling phase the newly extracted context vectors are compared with the code-book entries using a minimum (Euclidean) distance based on the histogram components.

The convergence properties of such relaxation operations have been studied before [46] and only guarantee convergence to local minima.

3.3 Performance Evaluation

For forestry inventory purposes, validation of the segmentation/labelling predictions are critical. Typically this step is ignored but some have reported uses of pixel-based differences between observed and predicted pixel values as one criterion [89, 3]. This criterion has severe drawbacks as it does not consider the fit between predicted and known annotated regions in terms of the region/shape information — the critical part of what humans are required to do in such tasks [59]. Consequently, we have used two types of measures: (1) a percentage correct detection score (PC), and (2), a region-based Percentage Overlap (PO) measure. The former simply determines if the centroid of the predicted class regions fall within a corresponding known region of the same class. The latter, being more exact, determines the degree to which the observed and predicted regions are identical in size and shape.

That is, for the latter measure, for all predicted regions R_j labelled i in the image, we compute:

$$PO(i) = \max_j \frac{R_i \wedge R_j}{R_i \vee R_j} \tag{3.19}$$

where $A \wedge B$ and $A \vee B$ refer to the set intersection and union of regions A, B, respectively. This provides an objective measure of how well the predicted annotated

regions fitted the observed ones. The HHMT parameters were then trained to maximize the *PO* scores, for each label over the training (mainly annotated) forestry image database.

3.4 Initial Comparison with Similar Methods

Although our main focus was on colour forestry images, for comparison purposes we first tested our algorithm on grey-level document images as used by similar models. In Fig. 3.3 we compare the HHMT algorithm with that of the TSMAP model [20] on the document images [20]. Note that TSMAP [5] uses a wavelet pyramid with 8 layers, 4 classes, and a 5x5 context kernel while our HHMT method uses a Haar pyramid of 4 layers, 4 classes, and 3x3 context kernel. Both methods are trained on the same 2 images [Fig. 3.3(a) and 3.3(b)]. To better demonstrate the results, (c) shows only a cropped part of one test image (All these three images are roughly the same size, approximately 700 x 1000). In detail, (d), (e), and (f) show the same part of the segmentation results using the TSMAP and HHMT methods, respectively. The TSMAP algorithm results in segmented images of 1/4 the scale of the original image (i.e., the layer next to the original image layer) while the HHMT method can have segmentation results at any given layer. So, to make the comparison more feasible, we present TSMAP's result in (d), and in (e) the HHMT's result in the same layer with (d), and in (f) HHMT's result at the original image layer. To facilitate comparison, (d), (e) and (f) are scaled up while keeping the same resolution ratio (e.g., (d) and (e) are the same in size and (f) is 2 times the size of (e) in both directions). As can be observed from this figure, the HHMT achieves slightly better segmentation results than TSMAP under visual inspection at the same layer (as shown in (d) and (e)), and much better and finer segmentation results at the original image layer. This initial test was performed to make sure that our algorithm could perform as well as what has already been published. However, the strength of the HHMT model lies in its additional characteristics as described above that allow us to apply it to quite complex coloured images like aerial forestry image data.

[5] This software is downloaded from the author's webpage http://dynamo.ecn.purdue.edu/ bouman/. We used the default parameter setting for TSMAP in the comparisons.

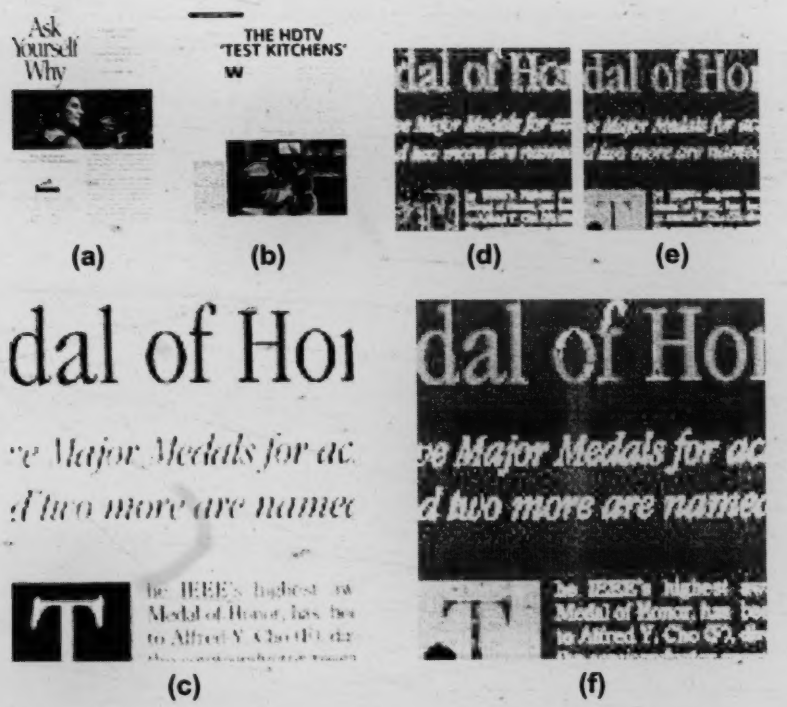


Figure 3.3: A comparison of performance of the TSMAP algorithm and the proposed HHMT method on grey-scale document images. (a) and (b) show example training images. (c) shows a cropped part of one of the test images. (d) shows performance of the TSMAP model while (e-f) show performance of the HHMT method (see text for more details).

3.5 Forestry Data Analysis

Our forestry image database consisted of 24 , 1024×1536 24–bit RGB images of experimental forestry plots, with an average of 64 trees per image. We evaluated our model on 6 of these images where we acquired expert annotation for $6 \times 64 = 384$ trees. These latter images were used for training and objective assessment of the model. The plots varied in terms of the degree of mixture densities of spruce and aspen — as shown in Fig. 3.4 and over several seasons: summer and fall and two weather conditions — sunny and cloudy.

These images are taken from the Alberta Research Council's experimental farm at Vegreville, Alberta. Throughout the simulations in this chapter, four different classes (scene labels) were used as annotation: Aspen, Spruce, Shadow and Ground. Due to our need to extract piecewise continuous canopy regions in this case, we have used Gaussian pyramids to define our hierarchical model with σ values of (2, 4, 8, 16

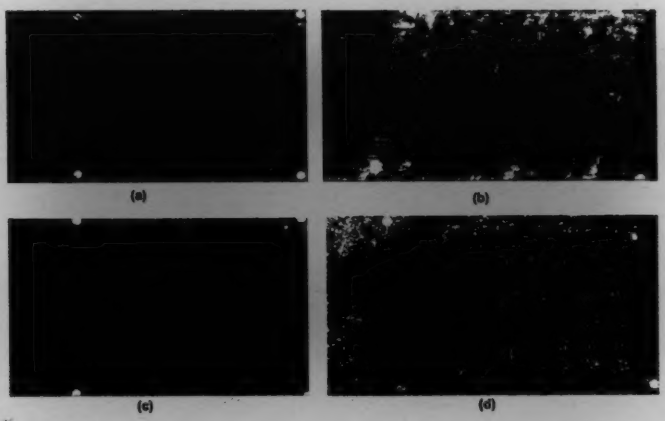


Figure 3.4: (a) Shows a simple regular spruce plot with some mixtures with aspen (left) on a sunny day. (b) Shows a complex plot consisting of a mixture of aspen and spruce on a sunny day. (c) Shows a simple regular spruce plot in cloudy day, and (d) shows an example of an early Fall image.

pixels,etc.). The image colour intensities are directly used as observed features.[6] For each such label, we have found that the parameters which offered the best *PO* scores occurred with the smallest kernels over all scales and largest or near largest number of scales, i.e., 3×3 and 6, respectively. Fig. 3.5 shows the difference between the performance of the Naive Bayes classifier alone, compared to the complete HMMT model with the 3×3 kernel and 6 layers.

For an objective assessment using the expert annotated images, we trained the HHMT model on a small segment (typically 300×300 in size) of an image, and then tested it on the complete image and/or other expert annotated images in the same season and climate. Table One summarizes a subset of performance in terms of the *PO* scores for each type of condition. Fig. 3.7 shows examples of how the HHMT model predicted regions and their annotations compared to experts. Although annotation for the more complex mixed stand regions is quite difficult even for experts, the results here are quite promising.

The HHMT model is a model for image segmentation, labelling and an image model-based recognition system in so far as the MAP value indicates the degree to which an image belongs to an image model (as shown in Fig. 3.6).

[6] It should be mentioned that the proposed method can be applied not only on raw multi-spectral images, but also on already processed multi-scaled image data (for example, Gabor-filtered images).

Plot type	Spruce	Aspin
Pure Spruce	80.60	—
Mixture	69.08	46.37
Fall mixture	75.60	50.24

Table 3.1: Shows Percentage Overlap (*PO*) scores for: column 1 - Spruce, column 2 - Aspen over the three different plot mixtures: pure spruce and mixture of Aspen and Spruce over one 1024 × 1536 representative test image per type.

3.6 Summary

The proposed approach differs from other current segmentation/labelling models in so far as image segmentation is not viewed as a prior process to region annotation and that the parameters that determine both processes can be learned from training examples. The specific algorithm explored here is very flexible and simple when compared to other current pyramid-based hierarchical Markovian models — as long as supervised learning, and training are available.

The proposed method can be applied not only to raw multi-spectral images, but also on already processed multi-scaled image data (for example, edge-encoded images). This is because the essence of the model is the segmentation and labelling of image data based upon the compatibilities of labels within and between scales, their relationship to observed values at each pixel, and evidence for such labels.

This type of model provides a number of image understanding operations in one integrated formulation. They are:

1. model-based hierarchical image segmentation,
2. image region labelling,
3. a form of structural image recognition, per se.

However, there are still a number of open questions that need pursuing. First, Gaussian mixture models have proven capable of coping with the structured image data we encountered in this forestry inventory application under the assumption that the outliers or noise do not degenerate the dominant principle components. To eliminate this assumption other mixture models should be examined. Second, the Tree Structured Bayesian Network (TSBN) [67] is another good choice for pre-class selection. In addition to the tree-structured decision making functionality that they both share, TSBNs can provide a soft-version of decision making that considers

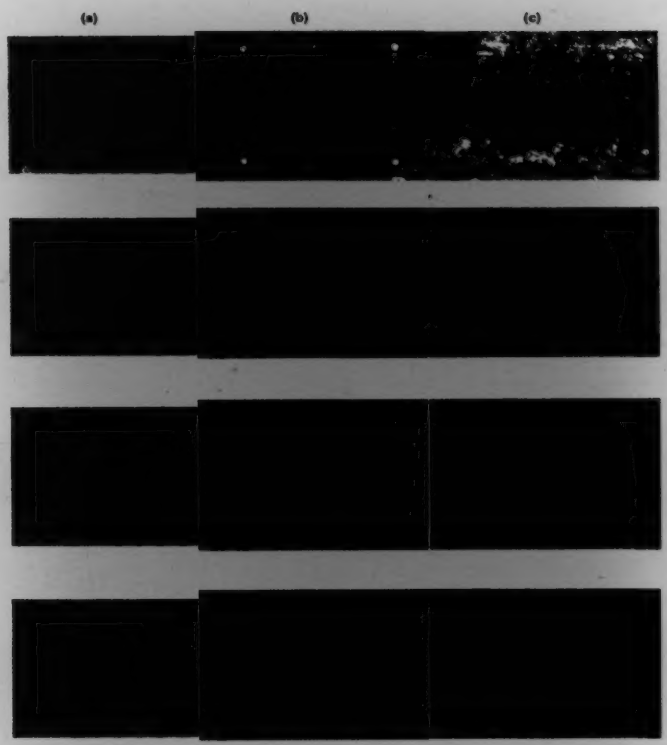
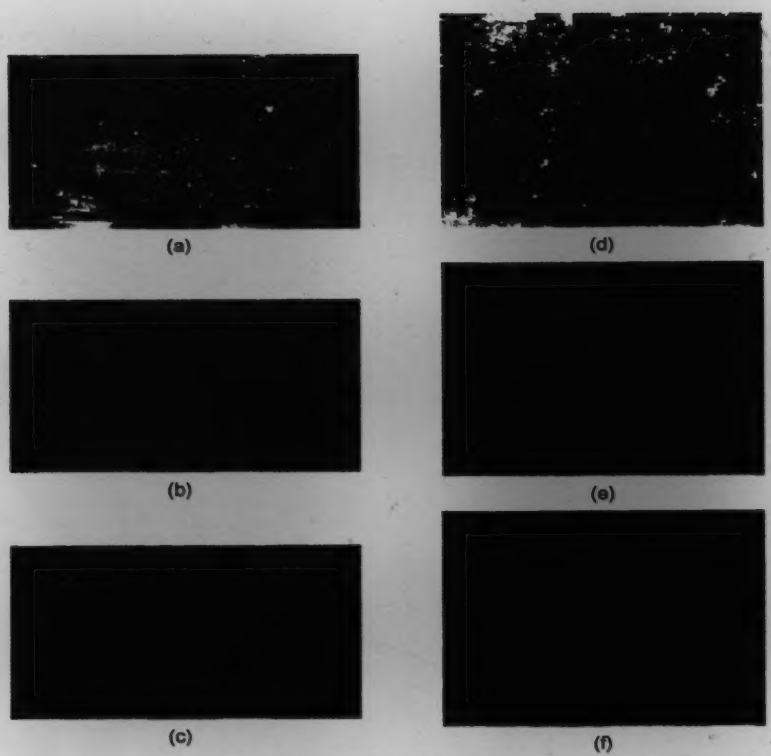


Figure 3.5: Shows performance on three types of forestry conditions: (a) Spruce only; (b) medium mixture of spruce and aspen; (c) significantly overlapping mixtures. Second row: classification by Naive Bayesian classifier. Third row: segmentation and labelling via the HHMT method. Forth row: expert annotation.

the complete decision path thus enabling users to choose the best possible path instead of falling into a sub-optimal decision path at an early stage. Third, in the current system, the bijective operators ran in fixed order: running downward from the uppermost layer to lowest, then upward till convergence reached. Other options exist including more local recursions.

In this HHMT model we used Gaussian pyramids. However, this may not always be optimal. For example, we conducted experiments using a Haar wavelet pyramid for the grey-scale document images [21], and modelled three wavelet coefficient bands as mixtures of multi-variate Gaussians (refer to Fig 3.3) with satisfactory results as well.



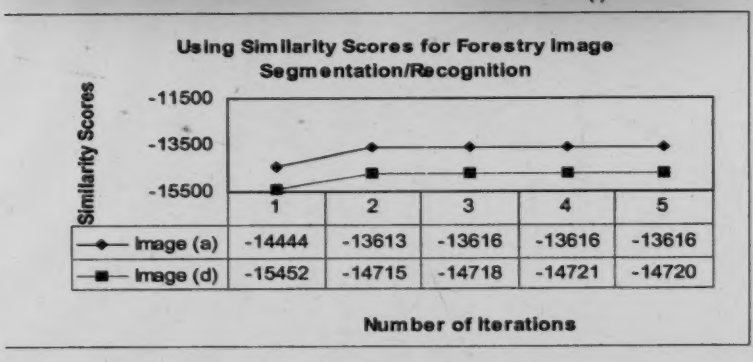


Figure 3.6: The HHMT model was estimated using image (a) and tested on images (a) and (d). We have used 4 classes corresponding to Aspen, Spruce, Shadows (both the shadow on the trees and on the ground) and Ground. Labelling results are shown using different colours. (b) and (e) show labelling after the first iteration (i.e., Naive Bayes Classifier) while (c) and (f) after five iterations. The bottom graph (g) shows the similarity scores (average log MAP probabilities per image) for the image data given the model as a function of the number of iterations of the algorithm.

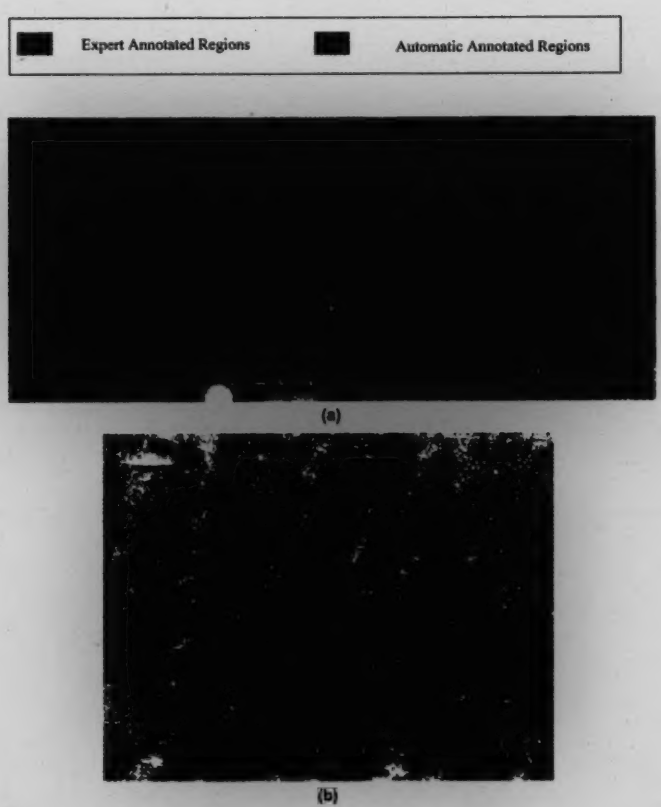


Figure 3.7: Shows more detailed examples of how the HHMT model fits expert annotation of spruce regions for the images in Figure 3.5.

Chapter 4

The Stereo Component

Assume a typical binocular dense stereo setting (e.g. [77]) where we are given two views of the same scene (left, right views, rectified to satisfy the epipolar constraint). The task is, with the reference view being the left view, to infer the disparity (or depth) of each pixel in that view. This could be easily extended to more general cases.

From the stereo viewpoint, the input images are a mixture of two types of smoothed patches: textured and untextured. Occlusion and discontinuities typically exist between neighbouring patches. Sensor noise, factors introduced via the image formation process, variations due to surface geometric/physical properties and lighting conditions, all contribute toward making stereo a difficult problem with many outliers existing in the data. Partly due to these factors, Gaussian-type models which lead to least square solutions can easily fail. Consequently we resort to robust statistics to cope with outliers as with line processing for containing discontinuous edges [11].

One major difficulty with stereo vision and also with other vision applications, is that typically hand-crafting of model parameters is required to adapt to unseen image pairs. In recent years there has been an attempt to overcome this problem by way of Bayesian analysis. For example, in medical image processing [44], Markov Chain Monte Carlo (MCMC) sampling techniques have been used with some success. Inspired by this, we propose a new stereo framework that contains two robust statistical functions controlled by hyper-parameters. Since these hyper-parameters are unknown, they are treated as random variables and then the task becomes that of learning the hyper-parameter values and the inference of the disparity map for specific input image pairs. Our framework is an extension of the linear state-space model (as with hidden Markov Models)[48] to two dimensional Markov networks. Unfortunately, the loopy network structure renders the exact inference (e.g., the Viterbi algorithm in HMMs) impossible, so loopy belief propagation (LBP) [91] [1] is used as the approximate inference engine. In addition, as an alternative to the EM algorithm for parameter estimation (e.g., the Baum-Welsh in HMMs), cycles of MCMC kernels [34] are employed to learn the hyper-parameters [57].

In the stereo vision literature, Barnard [6] proposed a simulated annealing al-

[1] We adopt LBP here for two reasons. First, LBP is rather efficient in comparison with the sampling approaches. Second, LBP has proven its effectiveness of approximation inference both in theory [91] and in practice [29]. This also raise an interesting issue of combining the sampling approach with deterministic approach.

gorithm to search for the optimal solution. Belhumeur [9] built a statistical model and solved the correspondence problem with a variant of dynamic programming. Recently, Sun et al. [79] also employed LBP to infer the optimal disparity map. However, the novel contributions of this work are to 1) explicitly address the stereo matching as a generative process, as shown in Fig. 4.1, 2) propose a Bayesian framework that naturally unifies the learning of both the disparity map and the hyper-parameters and 3) propose a new method to estimate the partition function based on a path sampling technique [31].

4.1 The stereo model

Define D be the predetermined range of possible disparity values. Let $i = 1, \cdots, n$ index a 2D lattice of image pixels, $y = \{y_i\}$ denote a 3D disparity space, with y_i a vector of length D, $d = \{d_i, d_i \in D\}$ denote the disparity map as a Markov random field (MRF). The proposed framework has two parts: the data model and the prior model where each is determined by a robust function. The data model $(y|d, \sigma_y, s_y)$ captures the interaction of the observation disparity space with the unknown disparity MRF while the prior model $(d|\sigma_d, s_d)$ defines the neighbouring interactions within the disparity map. $\theta = \{\sigma_y, s_y, \sigma_d, s_d\}$ are the hyper-parameters that control the two robust functions. In this chapter we derive a search algorithm using sampling techniques for learning the hyper-parameters, use the LBP inference engine to infer the optimal d.

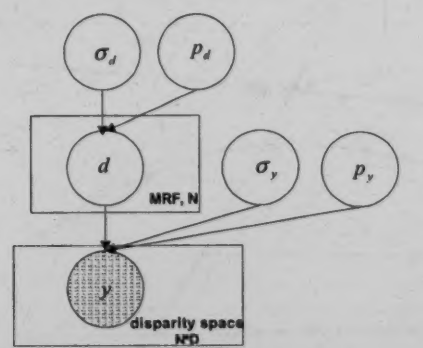


Figure 4.1: The proposed stereo model. The grey circle denotes the observable variable, y, corresponding to the disparity space computed from the left and right images and is the same size of the Markov Random Field (MRF), d, with each site containing a vector of size D. The white circles correspond to unobservable variables. The node d represents the disparity map. The σs refer to scale parameters and the ss refer to shape parameters.

Because of the uncertainty of θ for different image pairs (see Fig. 4.1) Bayesian analysis treats θ as unknown and assigns prior densities to θ. Since we can establish the likelihood $p(y|d,\theta)$, the priors $p(d|\theta)$ and $p(\theta)$, we can define the posterior as:

$$p(d,\theta|y) \propto p(y|d,\theta)\ p(d|\theta)\ p(\theta).$$

Optimization problem III: Depth from Stereo: The task is then two-fold. First, we need to infer the MAP disparity map d^* from:

$$d^* = \arg\max_d p(d|\theta^*, y)$$

where θ^* denotes the optimal estimate. LBP has been theoretically connected to the fixed points of Bethe free energy which is well-studied in statistical physics [91], and has success in vision [79]. Second, we also need to derive the adaptive estimation of hyper-parameters, θ,

$$\theta^* = \int_d \int_y \theta p(\theta) p(d,y|\theta) \mathbf{d}d\mathbf{d}y$$

with cycles of MCMC kernels [34], which draw dependent samples from the posterior of θ. This way we can evaluate the whole posterior distribution which incorporates our uncertainty of θ expressed in the prior, $p(\theta)$.

4.1.1 The Data and Prior Models

In this section we consider the data (likelihood) and prior models as a common unified function form[44]:

$$p(x|\sigma,s) = \frac{1}{\sigma^n\ z(s)}\ \exp\left\{-\frac{1}{s}\ u(x|\sigma,s)\right\}$$

where

$$u(x|\sigma,s) = \sum_i\ \rho(\frac{x_i}{\sigma},s)$$

is the energy function. x represents a random field, $\rho(\cdot,\cdot)$ is the potential function with scale parameter $\sigma \in (0,\infty)$ and shape parameter $s \in (0,2]$.

One reason for choosing this function is that the potential function $\rho(\cdot,\cdot)$ unifies many existing function forms, both convex and non-convex, into one general representation [44]. In particular, it includes the Generalized Gaussian distribution

$$\rho(\frac{x_i}{\sigma},s) = |x_i/\sigma|^s \tag{4.1}$$

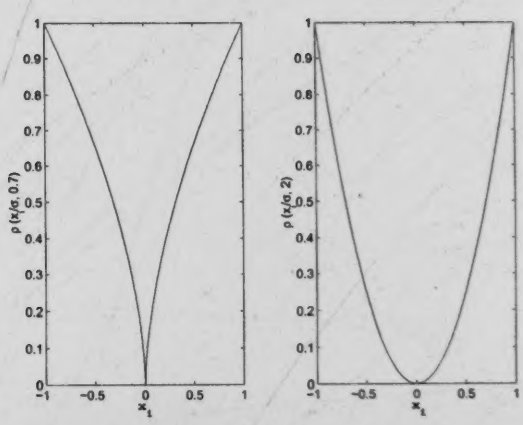


Figure 4.2: Examples of the potential function of the Generalized Gaussian distribution. Left: non-convex function when taking the shape value $s = 0.7$; Right: convex Gaussian function with the shape value $s = 2$ and the scale value $\sigma = 1$ for both cases.

and when $s = 2$, the Gaussian distribution:

$$\rho(\frac{x_i}{\sigma}, 2) = (x_i/\sigma)^2. \tag{4.2}$$

As shown in Fig. 4.2, the Generalized Gaussian density function covers a spectrum of density functions with the shape varying significantly from a more edge-preserving non-convex function with $s = 1$ (also called Laplace or double exponential function), to a smoother convex function with $s = 2$ (Gaussian density function).

Here we model the probability of the disparity space (the data model) given the disparity map (related to the hyper-parameters) as:

$$p(y|d, \sigma_y, s_y) = \frac{1}{\sigma_y^n} \exp\left\{-\frac{u(y|\sigma_y, s_y)}{s_y}\right\} \tag{4.3}$$

where σ_y corresponds to the scale parameter and s_y corresponds to the shape parameter for the data model.

Similarly, the Gibbs prior $p(d|\theta)$ of the MRF is defined by

$$p(d|\sigma_d, s_d) = \frac{1}{z(s_d)\ \sigma_d^n} \exp\left\{-\frac{u(d|\sigma_d, s_d)}{s_d}\right\} \tag{4.4}$$

where the energy function is slightly different due to the MRF:

$$u(d|\sigma_d, s_d) = \sum_{j \sim k} \rho(\frac{d_j - d_k}{\sigma_d}, s_d) \tag{4.5}$$

for $\{j, k\} \in n$ corresponding to all possible configurations of neighboring pixels, j and k. Notice that $z(s_d)$ in Eq. (4.4) is the partition function (normalizing constant) which is, in general, computationally intractable [48].

4.2 The Inference and the Learning Tasks

We consider the Generalized Gaussian function (see Eq. (4.1) in the appendix), and from the proposed generative model (Fig. 4.1), form the parameter space ($\theta = \{\sigma_y, \sigma_d, s_y, s_d\}$) MCMC kernels are adopted to estimate the two scale parameters, σ_y and σ_d and the corresponding shape parameters, s_y and s_d.

4.2.1 LBP for Approximate Inference of d

Here we describe how to apply LBP approximation in this setting. First we derive the full conditional probability over all MRF sites $d_i \in d$, and assume conditional independence of the nodes y_i given d_i:

$$\begin{aligned} p(d|y,\theta) &\propto p(y|d)\, p(d|\theta) \\ &= \prod_i p((y_i|d_i) \prod_{j\in\partial i} \psi_c(d_i, d_j) \end{aligned}$$

where $\psi_c(d_i, d_j)$ is the potential function modelling the interaction of d_i with its neighbouring node d_j. The belief for MRF, d, at node, i, is

$$b(d_i) \propto p(y_i|d_i) \prod_{j\in\partial i} m_i^j(d_i) \tag{4.6}$$

where the message update rules are

$$m_i^j(d_i) \propto \sum_{d_j} \psi_c(d_i, d_j) p(y_j|d_j) \prod_{k\in\partial j\setminus i} m_j^k(d_j).$$

After computing the belief at site i, we have

$$\begin{aligned} b(d_i) &\propto \sum_{1,..,i-1,i+1,..,n} \prod_j p(y_j|d_j) \prod_{k\in\partial j} \psi_c(d_j|d_k) \\ &\propto p(d_i|y_i,\theta). \end{aligned}$$

This implies that in this MRF network, the single belief $b(d_i)$ approximates the marginal probability $p(d_i|y_i,\theta)$ and in networks without loops, the beliefs are equal to the exact marginal probabilities [91].

Finally, it is easy to show that the joint probabilities over the hidden MRF d can be factorized as the product of beliefs over d

$$\begin{aligned} p(d|y,\theta) &\propto \prod_i p(y_i|d_i) \prod_{j\in\partial i} m_i^j(d_i) \\ &\propto \prod_i b(d_i). \end{aligned}$$

4.2.2 Updating σ_y and σ_d

Because there is no prior knowledge of σ_y, we assume a non-informative prior [30] [2], and according to the graphical model in Fig. 4.1, derive the full conditional probability of σ_y as inverse chi-square [30]:

$$\begin{aligned} p(\sigma_y^2|\cdot) &\propto p(\sigma_y^2)\, p(y|d,\sigma_y,s_y) \\ &\propto \frac{1}{\sigma_y^n}\exp\left\{-\frac{1}{s_y}\sum_i \rho(\frac{y_i}{\sigma_y},s_y)\right\}. \end{aligned}$$

σ_y is updated as

$$(\sigma_y^{s_y}|\cdot) \propto Inv-\chi^2(n,v) \tag{4.7}$$

(see Appendix C.2 for a detailed derivation)and

$$v=\frac{\sum_i |y_i|^{s_y}}{n}.$$

σ_d is updated similarly to σ_y, with

$$(\sigma_d^{s_d}|\cdot) \propto Inv-\chi^2(n,v) \tag{4.8}$$

and

$$v=\frac{\sum_{j\sim k}|d_j-d_k|^{s_d}}{2\,n}.$$

4.2.3 Updating s_y

With no prior knowledge of s_y, we assume a uniform prior distribution over s_y and derive the full conditional probability of s_y as

$$\begin{aligned} p(s_y|\cdot) &\propto p(s_y)\, p(y|d,\sigma_y,s_y) \\ &\propto \frac{1}{\sigma_y^n}\exp\left\{-\frac{1}{s_y}\sum_i \rho(\frac{y_i}{\sigma_y},s_y)\right\}. \end{aligned}$$

To make sure that s_y is drawn from the $(0,2]$ interval, we use the Metropolis sampling algorithm [34] with the proposal distribution $s_y^* \sim U[s_y^{t-1}-c, s_y^{t-1}+c]$. At step t, the candidate s_y^* is randomly accepted with probability

$$r=\min\{1,\alpha(s_y^{t-1},s_y^*)\} \tag{4.9}$$

[2] Non-informative prior allocates equal weights to all possible hypothesis in the parameter space.

where

$$\begin{aligned}\alpha(s_y^{t-1}, s_y^*) &= \frac{p(s_y^*|\cdot)}{p(s_y^{t-1}|\cdot)} \\ &= \exp\left\{-\left(\frac{\sum_i \rho(\frac{y_i}{\sigma_y}, s_y^*)}{s_y^*} - \frac{\sum_i \rho(\frac{y_i}{\sigma_y}, s_y^{t-1})}{s_y^{t-1}}\right)\right\}.\end{aligned}$$

4.2.4 Updating s_d

Since d is a MRF, updating the hyper-parameter s_d is more difficult due to the existence of the partition function. That is, the full conditional probability of s_d is

$$\begin{aligned}p(s_d|\cdot) &\propto p(s_d)\, p(d|\sigma_d, s_d) \\ &\propto \frac{\exp\{-\frac{1}{s_d}\sum_{j\sim k}\rho(\frac{d_j - d_k}{\sigma_d}, s_d)\}}{z(s_d)\,\sigma_d^n}.\end{aligned}$$

Therefore, to compute the full conditional probability of s_d we need to compute the partition function $z(s_d)$, at least up to some constant of proportionality and to sample $p(s_d|\cdot)$ we use the Metropolis algorithm. At step t, the candidate s_d^* is randomly accepted with probability

$$r = \min\{1, \alpha(s_d^{t-1}, s_d^*)\} \tag{4.10}$$

where

$$\alpha(s_d^{t-1}, s_d^*) = \frac{z(s_d^{t-1})}{z(s_d^*)} \exp\left\{-\left(\frac{\sum_{j\sim k}\rho(\frac{d_j - d_k}{\sigma_d}, s_d^*)}{s_d^*} \frac{\sum_{j\sim k}\rho(\frac{d_j - d_k}{\sigma_d}, s_d^{t-1})}{s_d^{t-1}}\right)\right\}.$$

The estimation of the partition function $z(s_d)$ is detailed in Appendix C.3. After applying the trapeziodal rule, the following approximation can be derived:

$$\alpha(s_d^{t-1}, s_d^*) = \exp\left\{-\left[\frac{1}{s_d^*}\sum_{j\sim k}\rho(\frac{y_j - y_k}{\sigma_d}, s_d^*) - \frac{1}{s_d^{t-1}}\sum_{j\sim k}\rho(\frac{y_j - y_k}{\sigma_d}, s_d^{t-1})\right]\right\}.$$

4.2.5 the Updating Strategy

An iterative procedure is employed to unify the LBP approximate inference in the inference step and the MCMC parameter learning in the estimation step using a fixed sampling scheme.

1. Initialize $(d^{(0)}, \theta^{(0)})$, set $t = 0$.
2. Iteration t of inference-estimation steps
   - Inference step: approximately inference of the disparity map $d^{(t)}$ by LBP
   - Estimation step: exploring the $\theta^{(t)}$ distribution by drawing N_t cycles of MCMC kernel samples. The Gibbs sampling is used to obtain *one cycle* of the kernels, as
     - Sample σ_y according to Eq. (4.7)
     - Sample σ_d according to Eq. (4.8)
     - Sample s_y according to Eq. (4.9)
     - Sample s_d according to Eq. (4.10).
3. $t \leftarrow t + 1$, goto 2.

In practice, the process converges after 2–3 iterations.

4.3 Experimental Results

Before treating the forestry images, for comparison purpose, experiments were conducted on the well-known testbed [77] with four stereo pairs: the "map" pair, the "sawtooth" pair, the "tsukuba" pair and the "venus" pair along with the evaluation methodology [77].

We first applied the Gaussian model to the testbed data as described in Section 4.2. The first two sets of data are shown (left view image only) in each first subplot of Fig. 4.3, and 4.4, with the inferred disparity map shown in the subplot (d) of Fig. 4.3 and (c) of Fig. 4.4, respectively. As expected, the Gaussian model tended to oversmooth the edges. We further broaden the parameters space by adopting the Generalized Gaussian model which has four hyper-parameters. Differences between the two models are illustrated in Fig. 4.4 (c) and (d) — the latter showing more sharp disparities at the edges and corners. Notice that the hyper-parameters always

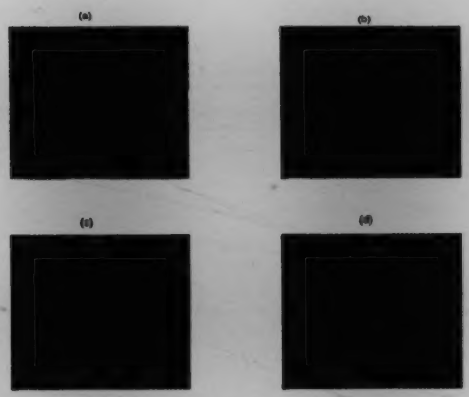


Figure 4.3: Experimental results for the "sawtooth" stereo pair. (a) shows the left view of the sawtooth stereo pair. When only employing the LBP algorithm for the inference of disparity map, (b) shows the resultant disparity map with fixed hyper-parameter values $(\sigma_d, \sigma_y, s_d, s_y)$ = (2,6,2,2), and (c) is the resultant disparity map with another fixed parameter values $(\sigma_d, \sigma_y, s_d, s_y)$ = (5,10,2,2). When the MCMC sampling techniques are employed, despite the different starting values as in (b) and (c), after a few iterations, hyper-parameters end up converging to fixed-point (0.01, 2.80, 0.73, 0.99), with the fixed-point disparity map inferred as shown in (d).

converged despite the different initial values. Fig. 4.6 shows the evaluation results on the testbed data. [3] Meanwhile, the disparity and error maps for all of the four testbed data are shown in Fig. 4.5. The corresponding estimated $(s_d, \sigma_d, s_y, \sigma_y)$ population mean values are: (0.77, 0.01, 1.09, 1.69) for "tsukuba", (0.81, 0.01, 1.26, 2.99) for "sawtooth", (0.73, 0.01, 0.99, 2.80) for "venus" and (0.55, 0.02, 1.17, 3.56) for "map".

Clearly the "map" pair benefits most from the Generalized Gaussian model (Fig. 4.4), with drastic reduction of error and the best ranking as shown in Fig. 4.6. The improvements to the other three image pairs was relatively modest. We postulate the following reasons for this. First, one set of hyper-parameters are used over whole image pixels, which indeed greatly reduces the computation effort while limiting the flexibility to adapt to different types of patches. Second, the current model is still a simplified version of the true model such that some important properties borne with the image formation process are ignored, such as occlusion, non-lambertian lighting conditions, etc. Also the LBP engine may also fail in cases where the loops in the MRF are strongly correlated [91].

[3] The "all." column is for the unoccluded regions, "untex" for the textureless regions and "disc" for the discontinuity regions as is the evaluation practice with this dataset [77]. The numbers are the evaluated errors via the aforementioned evaluation methodology, and the numbers in the brackets are the rankings in a current comparison table that is accessible via http://bj.middlebury.edu/~schar/stereo/web/results.php.

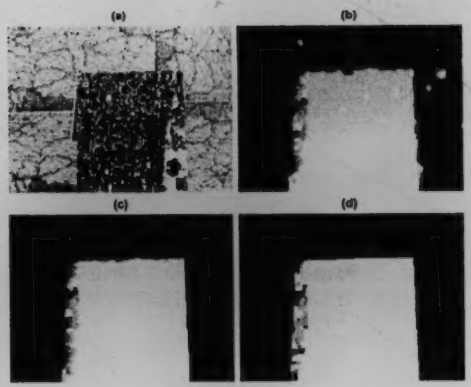


Figure 4.4: Experiments on the "map" stereo pair. (a) shows the left view of the stereo pair; (b) shows the disparity map with fixed parameter values $(\sigma_d, \sigma_y, s_d, s_y) = (2,6, 2, 2)$; (c) is the inferred disparity map by learning the scale parameters for Gaussian functions; (d) is the inferred disparity map by learning both the scale and the shape parameters for Generalized Gaussian functions.

For a final test, we created a small set of synthetic forest image pairs from sparse to dense with varying degrees of overlap, and applied the Generalized Gaussian model. Here we show one of the results in Fig. 4.7. Following the convention of the testbed data, we define the left view as the reference view, even though the data consisted of two images — a left and right view. We chose the disparity range as [0...12] and scaled it by 16 to form the grey-scale disparity map. The initial and the estimated $(p_d, \sigma_d, p_y, \sigma_y)$ hyper-parameter values are $(1, 1, 1, 6)$ and $(1.26, 0.02, 0.80, 1.46)$ respectively. The RMS [4] error (calculated only on the tree-related regions) of the final predicted disparity map was 0.78. Notice that these RMS errors are different from the aforementioned "bad pixels" evaluation errors used in Fig. 4.6.

In practice, we observed that 80 iterations of the LBP algorithm was enough to ensure the convergence, which took around one to five minutes, depending on the size and the maximum depth of the input images. Also, it took several minutes for the sampling algorithm to adapt θ values. Typically we run this twice.[5] So in total, the average running time was around ten minutes.

[4] The Root Mean Square (RMS) error is defined as: $\left[\frac{1}{n}\sum_{x,y}\text{diff}(x,y)^2\right]^{\frac{1}{2}}$ where diff$(\cdot)$ is the disparity differences and n is the number of pixels.

[5] In practice, we observed this is enough to guarantee the convergence of both θ and d.

Figure 4.5: The first column shows the resultant disparity maps for the four image pairs using the Generalized Gaussian model, and the second column shows the corresponding predicted disparity error compared to the ground truth. Here black pixels are counted as errors while gray pixels are in occluded patches and not counted.

4.4 Discussion

In this work we explored a novel perspective of stereo computation involving combining MCMC sampling algorithms with Bayesian inference techniques. Even with simple Gaussian and Generalized Gaussian models we are able to derive optimal hyper-parameter values that result in a generative framework which performs relatively well on a known testbed.

The proposed generative model is very flexible. One can have many ways to extend it for better performance. One natural direction is to relax the "tying" of θ. For example, pixels with similar (inferred) depth value can be spatially clustered together, then θ is tied only within clusters. This strategy is in line with the Swendsen-Wang type block-sampling techniques [80]. Besides, we can address the discontinuities/occlusions explicitly within the framework.

Algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untex.	disc.	all	untex.	disc.	all	untex.	disc.	all	disc.
Generalized Gaussian Model	3.65 (12)	4.24 (14)	15.33 (17)	1.23 (7)	0.41 (9)	7.91 (10)	3.41 (15)	5.80 (16)	17.40 (14)	0.10 (1)	1.33 (1)
Gaussian Model	7.02 (19)	9.52 (20)	19.97 (18)	3.14 (17)	2.09 (16)	22.38 (20)	5.34 (18)	8.24 (19)	34.82 (21)	4.94 (22)	25.33 (21)

Figure 4.6: The first row shows the evaluation errors for the four image pairs using the Generalized Gaussian model. The second row is the corresponding errors using the simple Gaussian model. Both are acquired with the aid of the online evaluation webpage that can be accessed via http://bj.middlebury.edu/~schar/stereo/web/eval.php. (See text for explanation of the ranking orders in the brackets.)

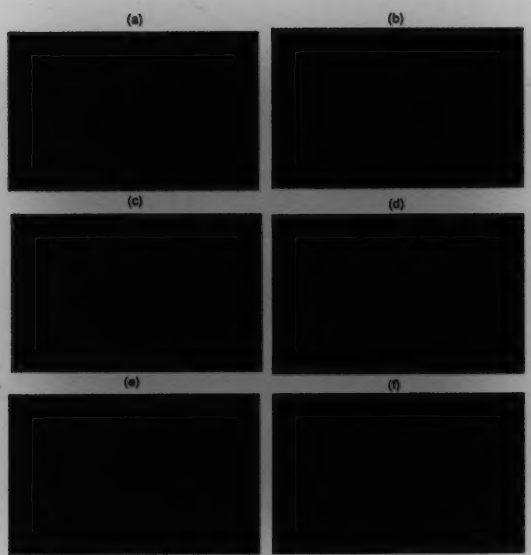


Figure 4.7: Experiment on one of the synthetic forest image pairs using the Generalized Gaussian model. (a),(b) show the left and side view of the trees — eight aspens and eight spruces. (c) is the disparity map of (a) at $t = 0$ step, and (d) is the disparity map at $t = 10$ step.

Chapter 5

The Tree Generation and Fitting

In this chapter, as shown in Fig. 5.1, we consider fitting super-ellipsoids [87][1] to the preprocessed data (or "sketches") which include the labelled tree regions with their corresponding depth map. This is a rather difficult problem, mainly due to

- the variability of the natural objects (trees, shadows, etc.) in the scene
- the noises introduced via the intermediate processes including the image formation process, the preprocessing processes, and so on.

Among these difficulties, the greatest we observed is the under-segmentation of tree regions. We observed that the labelled tree region often contains more than one tree, and far less the converse case when the region is over-segmented. Another reason of focus on the under-segmented case is the fact that the user of our system – the forestry industry cares more about the dominant trees which our segmenter rarely over-segmented in the simulations. Here we thus focus our attention on the under-segmentation case. In the future, it is straightforward to extend our algorithm to the over-segmentation case by traversing more state-spaces [2] in the Jump-Diffusion paradigm, as will be clear in later sections.

Optimization problem IV: 3D Trees Fitting: By assuming *iid* distribution over the labelled tree regions, we are ready to define the problem. Given the labelled tree region, the 3D tree fitting module needs to solve the optimization problems of

model choice: Decide on the number of 3D trees

parameter Estimation: estimate the parameter set for these 3D trees

inference: Infer the optimal scene geometry for this region.

Obviously the Bayesian paradigm stands out in this situation. By treating the problem in term of a Bayesian formulation, as schematically shown in Fig. 5.2 and 5.3, we are approximating the by exploring the posterior distribution of the unknown quantities, rather than estimating the single optimal point by the MAP algorithm which is less robust and will easily lead to overfitting [75].

Dellaert et al. [24] studied structure from motion using a version of the stochastic Expectation Maximization (SEM) algorithm [19] which is quite similar to our approach. The major difference is that they were trying to approximate the E step

[1] The super-ellipsoid is adopted here to approximate the geometry of 3D tree.
[2] Each state-space contains a different number of 3D trees for this labelled tree region.

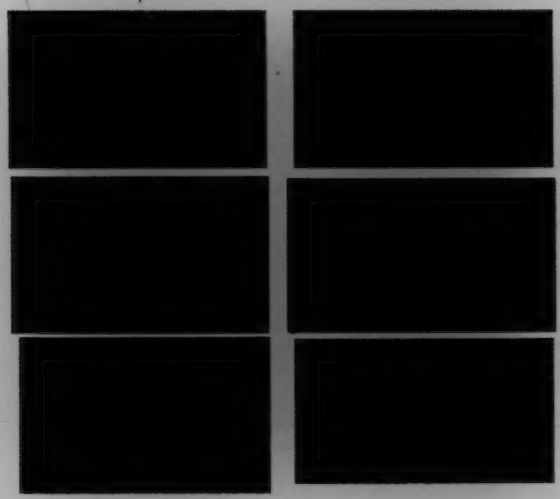


Figure 5.1: This example shows the typical results of the 3D tree fitting module. Top left is the left view of the input image pairs that contains two tree species — aspen and spruce. The bottom left shows the side view of the input scene geometry and the bottom right show the corresponding preprocessed data. This is the grey-scale depth map where the lighter the pixel value, the closer the object at that pixel to the viewer (camera). The 3D tree fitting module takes the preprocessed data and infers the latent 3D scene geometry, which is rendered back to 2D image with coloured texture mapped for different tree species, as shown in the top right.

by Monte Carlo sampling, while we propose to traverse the state-spaces by a Jump-Diffusion MCMC[3] (JDMCMC) sampler while the SEM algorithm is embedded inside the moves. The version of JD–MCMC we adopted here are closely related to the Data-Driven MCMC (DD–MCMC)[82] sampling algorithm proposed by Zhu et al.. We differ with their approach by the SEM algorithm adopted inside the sampler. Minka et al. [73], Beal et al. [7] proposed to solve data fusion by learning a Bayes net, which is similar to the graphical model we used here. They focused mainly on the parameter estimation and inference problem for the Bayes net. Our approach is also new to the image-based forestry inventory literature.

5.1 Problem Formulation and Notation

Define the set $\{l \in [1, L]\}$ indexing the labelled tree regions,[4] with the label set $\{t \in [1, T]\}$. Let us focus on segment l, and assume that all trees in this segment are of the same species, t. Fig. 5.2 shows the main graphical model of the proposed ap-

[3]The Jump process is to traverse the state-space by splitting or merging existing 3D trees. The Diffusion process is to make small perturbation/adaption within current state-space

[4]A labelled tree region is also called a segment in this paper.

proach, and on the right part of this figure, a zoomed-in view of the graphical model embedded in node x_l. In this Bayesian paradigm all unknown quantities (parameter nodes, latent variable nodes) are treated as random variables and are denoted as white nodes. In contrast, observed data are denote as grey nodes. Also, round nodes are for continuous and square nodes for discrete variables. The round node with a central dot [5] denotes a pre-fixed variable that comes from prior knowledge. In the main graphical model, node x_l denotes the latent 3D scene geometry and K denotes the number of trees, for segment l. Correspondingly, let $X = \{x_l; l \in [1, L]\}$ and $\{K\}_{l=1}^{L}$ be the latent geometry and tree counter[6] over all segments of the scene. The radial node attached to x_l contains the Mixture of Gaussian (MoG) parameters $\overline{\theta}_t = \{\overline{\mu}_t, \overline{\sigma}_t\}_{j=1}^{J}$ that encodes the prior knowledge of the tree species, t, and can be learned from training examples. In the zoom-in view on the right, there are K *iid* 3D trees inside node x_l. The associated parameters for each tree k ($\in [1, K]$) are $\theta_{k,l}$, with $\theta = \{\theta_{k,l}, k \in [1, K], l \in [1, L]\}$ as the parameter set and $\theta_{k,l}$ as the parameter for the kth 3D tree in the lth segment.

In the graphical model, instead of representing the preprocessed data directly, let these grey nodes denote the error measurements between the preprocessed data and those from the inferred scene geometry x_l. These errors can be computed once we have x_l. In turn, we will continue to call these error measurements "observed data". The three grey nodes, in order from left to right, denote observed data of the labelled regions from the left and the right views,[7] and the depth map from the left view, respectively. Accordingly, let $y_l = \{y_{l,lft}, y_{l,rgt}, y_{l,dep}\}$ denote the set of observed data over these grey nodes for segment l, with $Y = \{y_l\}$. Fig. 5.3 shows a simplified case of this figure when we restrict one 3D tree within the segment l. Then the two graphical models can be nicely integrated into one.

Assuming that the segments $[1, \ldots, L]$ are independent, the joint probability distribution function (pdf) is thus factorized

$$p(X, Y, \theta, \{K\}|M) = \prod_l p(y_l|x_l, \theta_l) p(\theta_l|K, M) p(K|M) \tag{5.1}$$

where $\{K\}$ is the set of tree counters, one per segment, and M denotes the proposed graphical model in Fig. 5.2. Let (K^*, θ_l^*, x_l^*) be the optimal solution for segment l.

[5] This node is called the radial node here.

[6] Each K counts the number of trees in a corresponding segment.

[7] Recall that the input data to our system consists of two images of the same scene, of slightly shifted views.

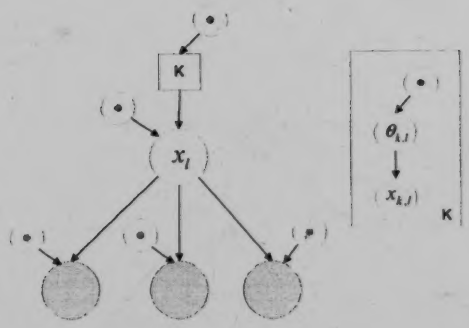


Figure 5.2: This figure contains a) the main graphical model of the proposed 3D tree fitting module, and b), on the right, the zoom-in view of the graphical model inside node S.

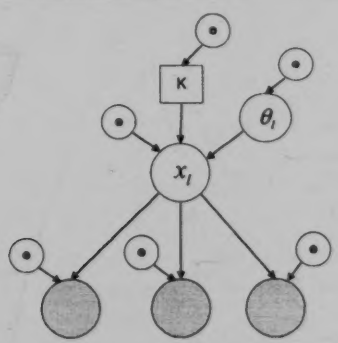


Figure 5.3: This shows a simple case of Fig. 5.2 when $K = 1$.

We are ready to formulate the problem

$$K^* = \sum_K K\, p(K|x_l, y_l, \theta_l, M) \tag{5.2}$$

$$\theta_l^* = \int_{\theta_l} \theta_l\, p(\theta_l|x_l, y_l, K, M)\mathrm{d}\theta_l \tag{5.3}$$

$$x_l^* = \arg\max_{x_l} p(x_l|y_l, \theta_l, K, M). \tag{5.4}$$

At first glance, it seems like an intractable problem with this Bayesian formulation, since besides the latent scene geometry, the set of parameters are to be estimated. Worst of all, the state-space $k \in K$ remains to be decided. It turns out that we can still find the approximate solution by a variant of the JD—MCMC [41] sampling technique to traverse possible state-space k, and a SEM algorithm to estimate the parameters θ and to infer the locally optimal scene geometry [8].

[8]SEM is guaranteed to converge to the local maxima point. Refer to Nielsen[66] for proof of its convergence property.

5.2 The Stochastic Expectation Maximization (SEM) Algorithm

In this section, we gradually develop the proposed SEM algorithm, by considering these cases in turn

Case 1 exactly one 3D tree per segment, $K = 1$

Case 2 K trees per segment, with K known

Case 3 K trees per segment, with K unknown

5.2.1 One 3D Tree per Segment

In this case, the parameters are simplified as $\theta = \{\theta_l; l = 1, \ldots, L\}$. Let $\theta_l = \mathcal{N}(\mu_l, \Sigma_l)$ with $\mathcal{N}(\mu, \Sigma)$ denoting the normal distribution where μ_l, Σ_l are the mean and covariance matrices,[9] respectively.

Given the objective function $l(\theta) = \log p(Y|\theta)$, the marginal log-likelihood function of the incomplete data, the Expectation Maximization (EM) algorithm can be interpreted as alternative maximization of the lower bound of $l(\theta)$[64]. Since the segments are independent, the expected log-likelihood function of the complete data is

$$Q(\theta) = \sum_{l=1}^{L} \int_{x_l} q(x_l) \log p(x_l, y_l|\theta_l) \mathbf{d}x_l \tag{5.5}$$

with $q(x_l) = p(x_l|y_l, \theta_l)$ being the estimation of the posterior pdf with the parameters fixed to the current ones. The complete data function can be further factorized

$$p(x_l, y_l|\theta_l) = p(y_l|x_l, \theta_l) p(x_l|\theta_l). \tag{5.6}$$

The lower bound is defined by

$$\mathcal{L}(q(X), \theta) = Q(\theta) - \sum_{l} q(x_l) \log q(x_l) \tag{5.7}$$

where $q(X) = \{q(x_l), l \in [1, L]\}$, we then have

E step $q(X)^* = \arg\max_{q(X)} \mathcal{L}(q(X), \theta)$

M step $\theta^* = \arg\max_{\theta} Q(\theta)$.

[9] It is a diagonal matrix under the assumption that the elements are independent.

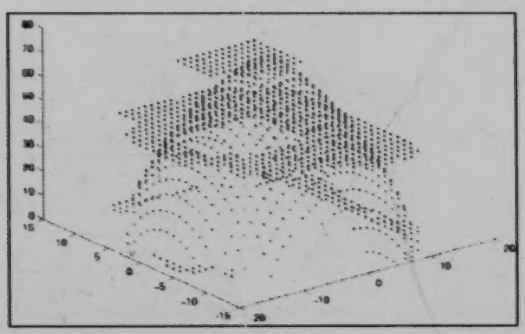


Figure 5.4: This figure shows the error measurements of the 3D tree volume (red curved surface) versus the corresponding depth map (blue surfaces).

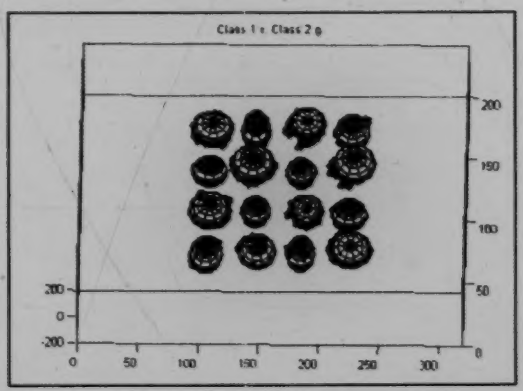


Figure 5.5: This figure shows a group of 3D super-ellipsoids with each sitting on top of a labelled 2D tree region. Notice in practice, the error measurements of the labelled regions are on 2D image plane with which the 3D trees rendered onto.

As mentioned earlier, we employ a variant of the SEM algorithm. For segment l, we draw S samples of x_l according to the $p(x_l|\theta_l)$. Each sample $s(\in \mathcal{S})$ is an empirical 3D assignment with weight

$$\widehat{h_l^s} = p(y_l^s|x_l^s, \theta_l^s).$$

Let us revisit Fig. 5.3. The three radial nodes associated with the grey nodes contain the parameters for data likelihood function $p(y_l|x_l, \theta_l)$, of the observed data (grey nodes). We render[10] the sample s as a 2D image, and define $A^s_{l,lft} = \{1..n^s_{l,lft}\}$ the pixel set it covered in the left view. Correspondingly $A_{l,lft} = \{1..n_{l,lft}\}$ is the pixel set covered by segment l in the left view. Similarly for the right view, they

[10]We do this by ray-tracing the 3D sample s back to 2D image, under the known motion parameters.

are $A^s_{l,rgt} = \{1..n^s_{l,rgt}\}$ and $A_{l,rgt} = \{1..n_{l,rgt}\}$. Since the three data likelihoods are independent, given the latent scene geometry x_l, they can be defined jointly as

$$\begin{aligned} \log \widehat{h}^s_l = const. - \sum_{i=1}^{n_l} \frac{(d_{dep,i} - x_{dep,i})^2}{2\sigma^2_{dep}} \\ - \frac{A_{l,lft} \backslash A^s_{l,lft} + A^s_{l,lft} \backslash A_{l,lft}}{2\sigma^2_{seg-lft}} \\ - \frac{A_{l,rgt} \backslash A^s_{l,rgt} + A^s_{l,rgt} \backslash A_{l,rgt}}{2\sigma^2_{seg-rgt}} \end{aligned} \tag{5.8}$$

where $\sigma_{seg-lft}$ and $\sigma_{seg-rgt}$, σ_{dep} are the parameters [11] for the observed data of the left and the right views, and of the depth map, respectively. $\cdot\backslash\cdot$ denotes the set-difference operation, which is deemed as a function here. *const.* contains some constant values and could be safely ignored in computation. The spatial and geometrical fitting problems described above, are illustrated in Fig. 5.4 and 5.5.

Therefore, the complete data log-likelihood can be approximated by

$$\widehat{Q}(\theta_l) = \sum_{s=1}^{S} \log \widehat{h}^s_l$$

$$\widehat{Q}(\theta) = \sum_{l=1}^{L} \widehat{Q}(\theta_l).$$

Here we skip the algorithm flowchart, since it can be derived as a special case of case 2.

5.2.2 Known Number of 3D Trees per Segment

It is a natural progression to extend the SEM algorithm to address the second case. Assume in segment l that the number of 3D trees is K, and the state-space is defined as $\Phi = \{\theta^K_l, K)$, where $\theta^K_l = \{\theta^K_{l,k}, k \in [1..K]\}$. The algorithm is described as

1. Initialization

   Set an initial θ value based on the empirical estimate of y_l.

2. Sampling step

   One set of samples is defined as the realization of K trees, for the current segment l. Draw S sets of such samples, according to θ_l.

[11] In Fig. 5.3, they correspond to the three aforementioned radial nodes.

3. E step

   Calculate

$$\widehat{Q}(\theta_{l,k}) = \sum_{s=1}^{S} \log \widehat{h}_{l,k}^{s}$$

   and

$$\widehat{Q}(\theta_l) = \sum_{k=1}^{K} \widehat{Q}(\theta_{l,k})$$

   where $\widehat{h}_{l,k}^{s} = p(y_{l,k}^{s} | x_{l,k}^{s}, \theta_l)$ are the soft weights, with

$$\begin{aligned}\log \widehat{h}_{l,k}^{s} = const. &- \sum_{i=1}^{n_{l,k}} \frac{(d_{dep,i} - x_{dep,i})^2}{2\sigma_{dep}^2} \\ &- \frac{A_{l,lft} \backslash A_{l,lft}^{s} + A_{l,lft}^{s} \backslash A_{l,lft} + \cap(A_{l,lft}, A_{l,lft}^{s})}{2\sigma_{seg-lft}^2} \\ &- \frac{A_{l,rgt} \backslash A_{l,rgt}^{s} + A_{l,rgt}^{s} \backslash A_{l,rgt} + \cap(A_{l,rgt}^{s}, A_{l,rgt})}{2\sigma_{seg-rgt}^2}\end{aligned}$$

   and where $n_{l,k}$ refers to the pixels in segment l that currently belong to the kth 3D tree. For current set of samples s, define $A_{l,lft}^{s}$ as the union of the pixel set the K trees covered in the left view. Similarly for right view is $A_{l,rgt}^{s}$. Define the $\cap(\cdot,\cdot)$ as the intersection operation function.

4. M step

   Update the parameters $\theta_{l,k}^{K} = (\mu_{l,k}, \Sigma_{l,k})$

$$\begin{aligned}\mu_{l,k} &= \frac{\sum_{s=1}^{S} \widehat{h}_{l,k}^{s} \cdot x_{l,k}^{s}}{\sum_{s=1}^{S} \widehat{h}_{l,k}^{s}} \\ \Sigma_{l,k} &= \frac{\sum_{s=1}^{S} \widehat{h}_{l,k}^{s} \cdot (x_{l,k}^{s})^2}{\sum_{s=1}^{S} \widehat{h}_{l,k}^{s}} - (u_{l,k}^{s})^2\end{aligned}$$

5. Iterate through step 2 to step 4 till converge for θ^{K}.

5.2.3 Unknown Number of 3D Trees per Segment

In this section, we investigate the case where one segment may contain an unknown number of 3D trees of the same type. This corresponds to a model choice problem [75]. It is well known that the ML framework will break down for this task: traversing the state-spaces of different dimensionality in deriving the optimal parameter set. Indeed maximizing the data likelihood biases solutions towards complicated

models. In this context it would end up with one pixel site fitted with a specific tree model – which is absurd.

On the other hand there are many constraints we can exploit (prior knowledge from training data, experts, etc.) regarding the relationships between heights, widths, and areas of given tree species. The prior knowledge can be encoded as the prior probability for the model parameters, θ. In the next section, we propose to use the JD–MCMC algorithm to determine the optimal state-space and estimate corresponding parameters from the joint posterior distribution of the state-space $\Phi = (\theta^K, K)$. To ensure computational efficiency, a sampling scheme is adopted which is in line with the DD–MCMC algorithm, as is detailed in the next section.

5.3 The Jump–Diffusion Sampling Paradigm

5.3.1 Problem Formulation

The prior knowledge about the state-space Φ for a given segment l, $p(\Phi)$[12], is defined by

$$p(\Phi) = p(K) \prod_{k=1}^{K} p(|R_k|)p(|\theta_k|)$$

where $p(|R_k|)$[13] is the prior for the 2D area formed via the rendering of the kth 3D tree onto the image plane. It is defined as $p(|R_k|) \sim \overline{\theta_t}$ which is a MoG for the estimated tree type $t \in \{1, \ldots, T\}$ that learned from the training data. $p(|\theta_k|) = \exp\{-\lambda|\theta_i|\}$ defines the prior for the number of parameters used for the kth tree, with λ a constant scalar. $p(K)$ is the prior for choosing the number of 3D trees to be K. In the current segment, it is defined as $p(K) \sim \mathcal{N}(\frac{A_l}{c_0 K}; \overline{\theta_t})$,[14] where A_l is the area of this segment, c_0 is a scaling constant.

$p(\Phi|x_l, y_l)$ defines the joint posterior pdf of being in the current state-space

$$p(\Phi|x_l, y_l) \quad = \quad \pi p(y_l|x_l, \Phi)p(x_l|\Phi)p(\Phi)$$

where π is the normalization factor.

Let Ω denote the solution space which consists of a union of possible state subspaces

$$\Omega = \cup_{n=1}^{N} \Phi_n. \tag{5.9}$$

[12]To simplify the notation, we sometimes omit the model term M with no confusion. As an example, $p(\Phi|M)$ is noted as $p(\Phi)$.

[13]We use R_k to refer to the kth 3D tree in the current segment and will extensively use it in the following derivations. This way, it is natural to define $x_l = \{R_k; k \in [1, K]\}$.

[14]Let $\mathcal{N}(\cdot;\cdot)$ or $\mathcal{N}(\cdot;\cdot,\cdot)$ denote the Normal or MoG distribution.

This implies that our algorithm actually scales linearly with the increasing number of state-spaces, as $N \to \infty$. Besides, $[1, N]$ is obviously a limited range in our situation, since it is impossible to have infinitely many trees in a segment with reasonable size.

5.3.2 the Proposed Algorithm

We apply the proposed algorithm to each of the segment $l \in [1, \ldots, L]$, sequentially.

1. Initialization:

   - Set $K^{(0)}$ to be any reasonable integer, for example, $K^{(0)} = 1$. Then apply the aforementioned stochastic EM algorithm to obtain $\Phi^{(0)} = \{\theta^{K,(0)}, K^{(0)}\}$.

2. For $i = 1$ to N

   - Sample $\nu \in U[0, 1]$ [15]
   - if $\nu \leq s_K$, then split move
   - else if $\nu \leq s_K + m_K$, then merge move
   - else the update move.

Given current state-space $\Phi = (\theta^K, K)$ at iteration i, we can choose from three type of moves: the update move with probability u_K, where K is unchanged and θ^K is updated; the merge move with probability m_K, where we jump to a new state-space with lower dimensionality, $K^{(i+1)} = K - 1$, with $\theta^{K-1,(i+1)}$ initialized and briefly updated, with the SEM algorithm; and finally, the split move with probability s_K, where we jump to a new state-space with higher dimensionality, $K^{(i+1)} = K + 1$, with $\theta^{K+1,(i+1)}$ initialized and briefly updated, with the SEM algorithm. According to Green [39], these move probabilities are defined as

$$s_k = c \min\left\{\frac{p(K+1)}{p(K)}, 1\right\} \tag{5.10}$$

$$m_K = c \min\left\{\frac{p(K)}{p(K+1)}, 1\right\} \tag{5.11}$$

$$u_K = 1 - s_K - m_K \tag{5.12}$$

where c is a scaling parameter to adjust the proportion of the jump move (split or merge move).

[15] $U[\cdot, \cdot]$ denotes the uniform distribution bounded within a range.

5.3.3 The Update Move

Within the state-space defined via $\Phi = \{\theta^K, K)$, the update move is to fit the parameters (θ^K) of K trees to the current segment, l, by the aforementioned SEM algorithm when K is fixed.

5.3.4 The Merge and Split Moves

In the literature, two popular approaches have been developed for the split/merge moves, among others. The first one is the Reversible-Jump MCMC (RJ–MCMC) [39] and the second one is the JD–MCMC paradigm. We adopt the second approach because the RJ–MCMC is inefficient for this specific application where we have strong prior constraints (statistical information about tree area, height, etc.), plus, the observed data contains distinct cues (the shape of nearby shadow regions, etc.) that could contribute to the data likelihood term after model adaptation. These could result in reducing the search space and speed up the algorithm greatly if an appropriate sampling strategy is adopted. As a variant of the JD–MCMC paradigm, the DD–MCMC is distinct for its efficiency when the data likelihood term can be efficiently learned from data, and can thus be put into the proposal distribution while still maintain the detailed balance. We follows this approach for our version of the JD–MCMC paradigm.

For segment l, assume we are now in the state-space $\Phi = \{\theta^K, K)$ and about to move to $\Phi' = \{\theta^{K+1}, K+1)$ $(\Phi \to d\Phi')$ via a split move. This move is accepted with probability

$$\alpha(\Phi \to d\Phi') = \min\left\{1, \frac{p(\Phi'|x_l, y_l)G(\Phi' \to d\Phi)d\Phi'}{p(\Phi|x_l, y_l)G(\Phi \to d\Phi')d\Phi}\right\} \tag{5.13}$$

where $p(\Phi'|x_l, y_l)$ and $p(\Phi|x_l, y_l)$ are the posterior probabilities. $G(\Phi \to d\Phi')$ and $G(\Phi' \to d\Phi)$ are the proposal probabilities for, respectively

- split move

$$\begin{aligned} G(\Phi \to d\Phi') =& q(\text{split})q(R_s)q(R_i, R_j|R_s) \\ & q(\theta_i|R_i, y_l)q(\theta_i|R_j, y_l)d\Phi' \end{aligned} \tag{5.14}$$

- merge move

$$\begin{aligned} G(\Phi' \to d\Phi) =& q(\text{merge})q(R_i)q(R_j|R_i) \\ & q(R_m|R_i, R_j)q(\theta_m|R_m, y_l)d\Phi. \end{aligned} \tag{5.15}$$

$G(\Phi \rightarrow d\Phi')$ could be interpreted as probabilistically describing how one series of actions are possible to happen. In detail: let us choose a split move with probability $q(\text{split})$; then choose a 3D tree R_s with probability $q(R_m)$; split R_s into R_i and R_j with probability $q(R_i, R_j|R_s)$. Finally, the last two terms, $q(\theta_i|R_i, x_l, y_l)$ and $q(\theta_i|R_j, x_l, y_l)$, are the data likelihood terms associated for the two newly split regions.

Similarly for $G(\Phi' \rightarrow d\Phi)$, we use the same protocal: we choose a merge move with probability $q(\text{merge})$; then tree i is chosen with probability $q(R_i)$, and its neighbouring tree R_j given R_i is chosen with probability $q(R_j|R_i)$; R_i and R_j are merged into R_m with probability $q(R_m|R_i, R_j)$, where $q(\theta_m|R_m, x_l, y_l)$ is the data likelihood term for merging the two regions R_i and R_j.

Notice since we always alternate between the state-spaces, the Jacobian terms $d\Phi$ and $d\Phi'$ are cancelled out . According to Bayes rule, we can obtain the two jump moves. First, the split move

$$\begin{aligned}
&\alpha(\Phi(K) \rightarrow d\Phi(K+1)) \\
&= \frac{q(merge)q(R_i)q(R_j|R_i)q(R_m|R_i,R_j)q(\theta_m|R_m,y_l)}{q(split)q(R_s)q(R_i,R_j|R_s)q(\theta_i|R_i,y_l)q(\theta_j|R_j,y_l)} \\
&\quad \frac{p(K+1)p(R_j)p(|\theta|)}{p(K)p(R_s)} \frac{p(x_l,y_l|\Phi(K+1))}{p(x_l,y_l|\Phi(K))}
\end{aligned} \tag{5.16}$$

Notice they are nicely factorized as a product of three ratios. The first ratio contains the ratio of two proposal probabilities, the second is the ratio for prior probabilities and the third is the ratio for data likelihood.

Similarly, we have the merge move which is the inverse of the split move $\alpha(\Phi(K) \rightarrow d\Phi(K-1)) = \frac{1}{\alpha(\Phi(K-1)\rightarrow d\Phi(K))}$.

5.4 Implementation and Experiments

We defer the detailed definition of $\theta_{l,k}$ until this section. According to the derivation in previous sections, it is clear that for segment l, $\theta_{l,k}$ are the parameters that determine the distribution function of the $k$the 3D tree. In practice, define, in 3D space, the x–z axes forms the ground plane and the y axis the height of the tree. A 3D tree is uniquely determined by the following parameters: the x, y, z position for the center of the tree bottom; the scales along x, y, z axes; one scalar for the rotation along the y axis; one scalar parameter for the super-ellipsoid equation. In our implementation, $\theta_{l,k}$ is defined as the multivariate normal distribution with an

Figure 5.6: A typical simulation run on the testbed data. Top left: the left view of the input image pair. Bottom left: the labelled regions. Bottom right: the inferred scene geometry using initial estimated parameters, with one tree per segment. Top right: the final result after running the proposed JD–MCMC algorithm to convergence.

Figure 5.7: Given one segment, started from two 3D trees, this shows the dynamics of the fitting process. Top left: the left view of the input image pair. Bottom left: Initially inferred scene geometry. Bottom right: result of the split move at the 5th iteration. Top right: result after 40 iterations.

8–dimensional mean vector and 8×8 diagonal covariant matrix.

In the simulations, for the proposed JD–MCMC paradigm, $N = 100$ is empirically sufficient to ensure convergence to a stable solution for a given segment. c is fixed to 0.5 to enforce the probability for a jump move (split or merge move) in the range of $[0.5, 1]$. The scaling factor c_0 is fixed to 1.5. The penalty constant λ is fixed to 0.2. For the measurement (observed data) related parameters, $\sigma_{seg-lft}$ and $\sigma_{seg-rgt}$ are set to 0.3, and σ_{dep} is set to 0.2. The tree species priors $\bar{\theta}$ are MoGs that learned from training images off-line.

As preliminary results, we test our algorithm on a set of synthesized forestry image data, with varying degrees of overlap for neighbouring trees. Preliminary results are obtained on this testbed, as shown in Fig. 5.6. The fitting dynamic of one segment is shown in Fig. 5.7.

5.5 Summary

In this chapter, within the context of building a (both 2D and 3D) data fusion module for a practical image understanding system, we propose a Bayesian paradigm to jointly solve the model choice, the estimation, and the inference problems. Encouraging results are obtained on synthesized image data.

Future work will involve more simulations on real image data. For further improvement, the proposed paradigm as shown in Fig. 5.2, is very flexible. Thus it can be extend to contain more sensor data by adding new measurement nodes. Besides, it is possible to extend to handle over-segmentation case.

Finally, to render the 3D scene geometry to a 2D image plane, a free ray-tracing software is utilized [1] that may later be replaced with our own ray-tracer for better control of the geometry checking.

Chapter 6

Experimental Results

The FORIUS system, when fully functional, requires the following components:

1. Fully calibrated [1] geo-referenced multi-spectra stereo image data
2. Ground-truth interpretation of training images consisting of representative stands, terrain
3. Camera-related information, for example, we need to know *apriori* the baseline b and focal length f to derive the depth map (z elevation map) from disparity map d [2]
4. Expert prior knowledge of the statistics of height, regional area, etc. of each tree type, of the plot of interest.

In addition to the experimental studies reported in the previous chapters, in this chapter we report additional results on a range of forestry image data.

- To evaluate the segmenter and annotator module, we used a set of top-view colour images of trees taken from a plantation of the Alberta Research Council (ARC), Canada, as shown in Chapter 3.6. We also experiment on a set of grey-scale airborne forest images from ARC, as shown in Sec. 6.1.
- We tested the FORIUS system on a set of stereo image pairs generated from a synthetic forest with varying resolutions, as detailed in Sec. 6.2.
- In Sec. 6.3, using a set of synthesized images with overlapping (among trees), we ran simulations to test the performance of our system in such under-segmentation scenarios.
- Finally, we evaluated our system on the colour airborne forestry images, from Timberline Inc., Alberta, Canada, in Sec. 6.4.

6.1 Data One: Evaluation of the Segmenter-Annotator

For this set of single view image data, we can only conduct experiments on the segmenter-annotator component. Future plans include replacing the stereo component with one that rely on computing the "taper functions". These functions are

[1] The image pair are aligned such that the epipolar constraint holds.

[2] According to the geometric model for stereo system, we know $z = fb/d$, refer to pp. 398 of [78] for more details.

used for estimating the tree heights based on the tree regions. This way the whole framework can be fully engaged in this task.

Currently two sets of such kind of image data are used for the simulations:

1. ALPAC [3] cut-plot image data, around 100 cm per pixel resolution, grey-scale images. (See (a), (c), and (e) of Fig. 6.1 as examples.)

2. Airborne image data, around 30 cm per pixel resolution, colour images. (See (a) of Fig. 6.2 as examples.)

The experiments on these two type of data, as shown in Fig. 6.1 and 6.2 are conducted while only the segmenter-annotator is employed. The results visually demonstrate the robustness of the proposed segmentation module, in situations when only the grey-scale image data are used (as in Fig. 6.1), or when the objects of the images are difficult to be discern even by human expert. According to the experiments on these two data sets, the segmenter-annotator performs best on multi-spectral image data with sufficiently high resolutions (for example, the 30cm per pixel resolution).

6.2 Data two: Evaluation of the Synthetic Forestry Images at different resolutions

Stereo image pairs with different resolutions are shown in Fig. 6.4 and Fig. 6.5. Five pairs of images are shown, where each row contains the left/right views of a plot with 16 trees (8 aspens and 8 spruces) of the same resolution, and each column contains the plot view with reduced resolution. These served as input images for this simulation. The segmenter-annotator is trained on a small portion of (a), as shown in Fig. 6.3, at each respective resolution. Notice the number of trees do not need to be known beforehand.

The simulation results are shown in Fig. 6.6 through to Fig. 6.11 at resolutions range from 20 cm per pixel to 100 cm per pixel. As a summary, Fig. 6.12 plots detection and labelling accuracy (intersection/union ratio or confidence) against image resolution (in the scale of centimeter per pixel), for different tree types.

It is evident from Fig. 6.12 that the predict (segmentation, stereo, and scene geometry) results degrade as the image resolutions goes down. In detail, the segmentation results is robust against the changes in image resolutions, and even at

[3] ALPAC is the abbreviation of Alberta-Pacific Forestry Industry Company, Alberta, Canada.

80cm/pixel, quite acceptable detection and labelling results were found. However, stereo performance decreased with decreases in image resolutions. The fusion of the two components delivered relatively good results but the trees in the 2D rendered image (of predicted scene geometry) are obviously larger then those in the input image [4] due to the errors in the stereo algorithm. It is clear that 20 cm/pixel image resolution is the best choice. The proposed system is reasonably robust till the image resolution goes coarser than 100 cm per pixel, when it encounters difficulties in the accurate estimation of the depth information.

6.3 Data Three: Evaluation of the Synthetic Forestry Images with Issue of Under-segmentation

We conducted a series of experiments on a set of the synthetic forestry images which was the same dataset as used in Sec.5.4. To demonstrate the robustness of the 3D fitting module on the under-segmented scenario, we set $K^{(0)}$ with different initial values [5]. From the results shown in Fig. 6.13 and 6.14, the 3D fitting module performs relatively well, with the results converged to the same scene geometry, despite the rather different starting positions of $K^{(0)}$.

6.4 Data Four: Evaluation of the Timberline Image Dataset

In this section, we experiment our FORIUS system with the Timberline forestry image pair.

The Timberline dataset consists of 14 airborne colour (RGB bands) images on various forestry plots [6] of the athabasca area, northern Alberta, Canada, with the resolution of around 30cm per pixel. Each image is geo-referenced, is of 15328×15328 in size, and has around 60% overlap with the consecutive images (since the images is taken sequentially during the flight). Unfortunately not all the image data are useful to us, due to either the stereo pair constraint, or the plot of interest. To be precise, since a single camera was employed for producing the image data, a large portion of the image data can not be used for lack of stereo correspondence. Also

[4] This trend becomes obvious when the image resolution is coarse than 80cm per pixel.

[5] See sec.5.3.2 for algorithm details.

[6] The images are taken by a fix-wing aircraft during 17:13 to 17:50pm, September 13th, 2003. the camera mounted was a Zeiss RMK Top 15 with a calibrated focal length of 153.371 mm. and the photo scale is around 1:20,000 (with the nominal elevation is at 12,290 feet above sea level).

a large portion of the image covers sites (lakes, ranches, etc.) are of less interest to us. We have to select from the overlapping parts some images of smaller size that can form stereo pairs and cover interesting forestry plots.

We conducted a series of experiments on this Timberline dataset. 6.15 shows such experiments with intermediate results for each modules. You may notice the poor performance of the stereo module. This is mostly due to the fact this Timberline image pairs are wide-baseline stereo,[7] which is different from the standard stereo setting that we assumed in our stereo module. However, in this scenario, the sketches acquired from the segmentation module help to maintain relatively good performance during the 3D fitting stage. In another word, the proposed FORIUS system perform rather robust given the degraded behavior of one of the components.

6.5 Summary

In this chapter, we have evaluated the performance of the proposed FORIUS system under different scenarios. From these experiments we conclude that the FORIUS system behaves rather robustly for various situations, given adequate data.

In controlled experiments (as in Sec. 6.3) where the camera geometry are known and the high resolution colour image pair obey the epipolar constraint, the FORIUS system infers the forest scene geometries that are visually very close to the ground truth. This is evidently shown in Fig. 6.14 where the 2D projection of the inferred scene geometries are visually compared with the ground truth from the same side view.

In the case (as shown in Sec. 6.4) where camera geometry are partially known, and even the known parameters are not accurate, and the image pair do not obey the epipolar constraint, amazingly, the FORIUS system can still work and produce plausible scene geometries, as shown in Fig. 6.15.

[7] The wide-baseline stereo refers to the situation when 1) the two cameras are very far apart, which causes problems in identifying correspondences versus occlusions, and 2) the two views do not strictly obey the epipolar constraint.

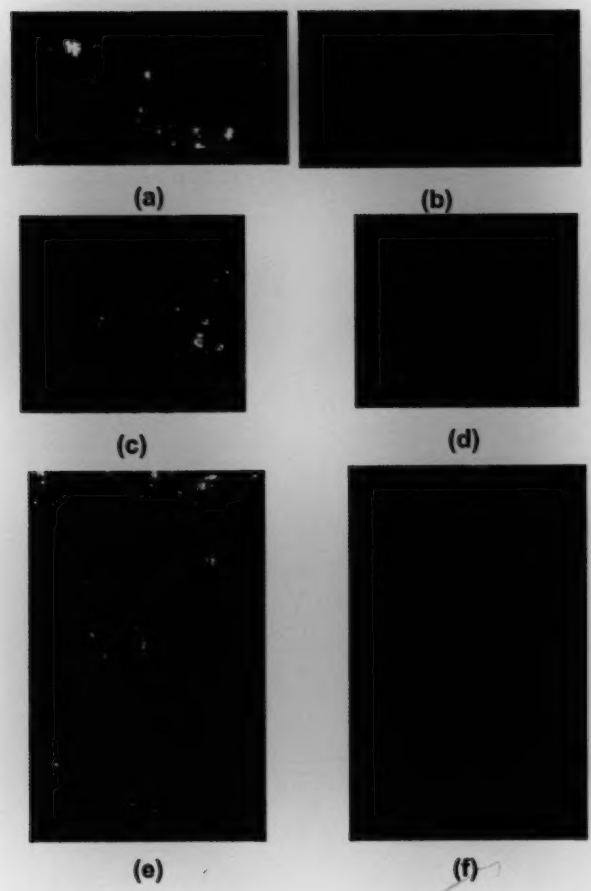


Figure 6.1: An example of the experiments on airborne grey-scale image data of around 100 cm per pixel. (a) The input cut-plot image. (b) The segmentation result of (a) with intersection/union confidence 0.5350 (includes the tree and the shadow classes). (c) The input cut-plot image. (d) The segmentation result trained on (a), with intersection/union confidence 0.3191 (includes the tree class and the shadow class). (e) The input cut-plot image. (f) segmentation result trained on (a), with intersection/union confidence 0.3653 (includes the tree class and the shadow class). Here we used the average colour for each segmented regions. The image consists of three classes, while green colour is for tree pixels, black colour is for shadows and brown colour is for the remaining pixels.

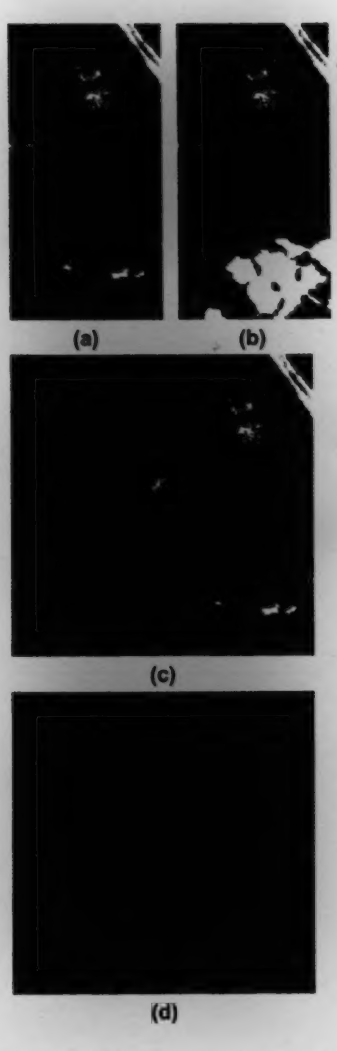


Figure 6.2: An example of the experiment on airborne colour data of about 30 cm per pixel. (a) the training image. (b) The expert labelling result. (c) The testing image. (d) The segmentation result with intersection/union confidence 0.3726 (includes two tree types only) or 0.2103 (includes two trees types and the shadows). The image consists of four classes. In the expert labelling image, the green and yellow colours are labelled to identify the pixels of the two tree types, respectively, blue colour is for shadow pixels and the remaining is left unmarked.

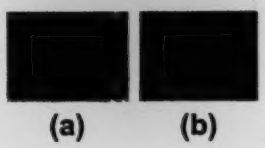


Figure 6.3: This shows the training image used with 20cm/pixel resolution. In expert labelling of this set of experiments, red colour is marked for aspen tree, green colour is for spruce tree, blue colour is for shadow and the remaining is left unmarked.

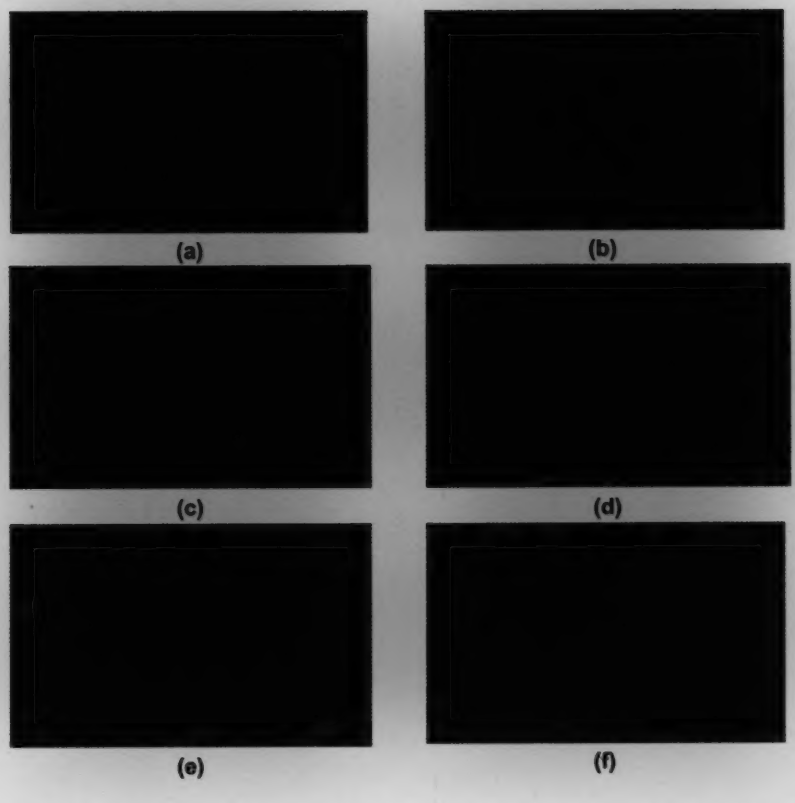


Figure 6.4: Plot of 16 trees (left two columns of spruce, right two of aspen) with equivalent image resolutions of (20,30,40) cms/pixel.

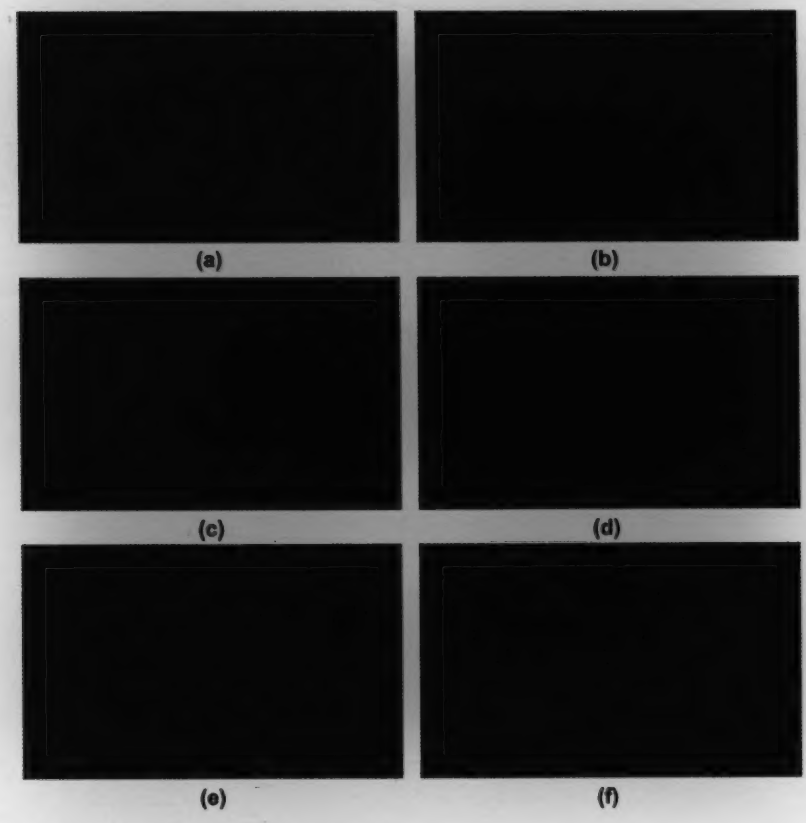


Figure 6.5: Plot of 16 tree (left two columns of spruce, right two of aspen) with equivalent image resolutions of (60,80,100) cms/pixel.

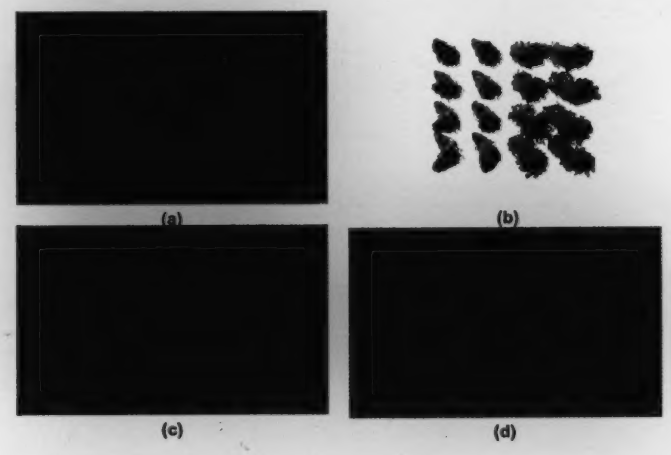


Figure 6.6: The simulation results for the resolution of 20cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result. (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

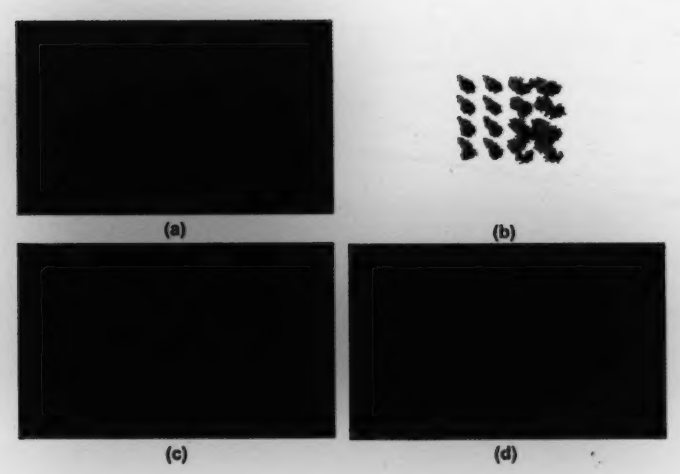


Figure 6.7: The simulation results for the resolution of 30cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

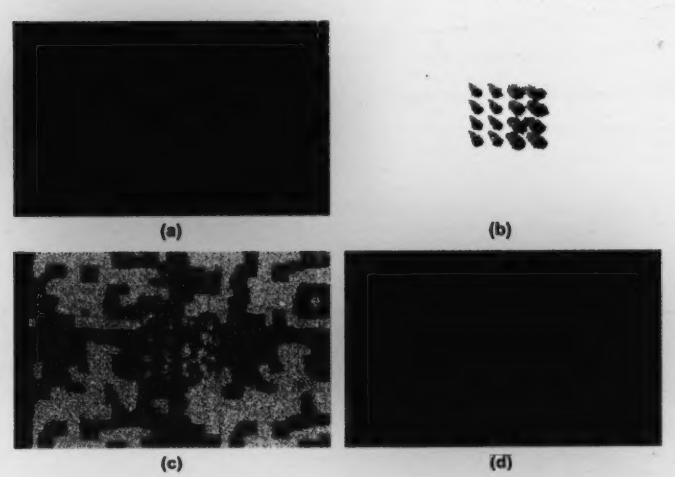


Figure 6.8: The simulation results for the resolution of 40cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

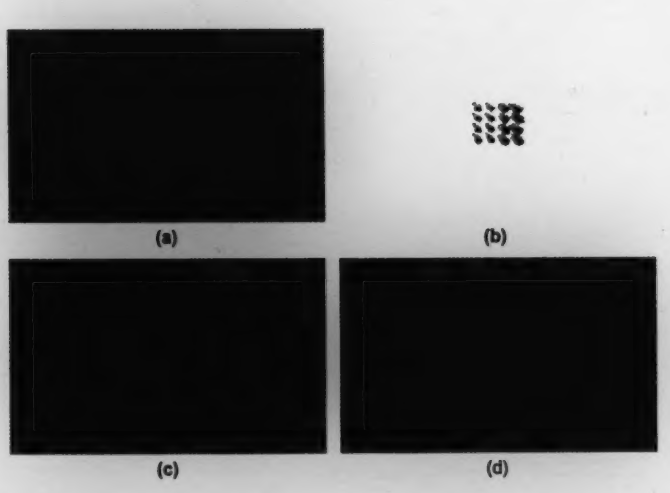


Figure 6.9: The simulation results for the resolution of 60cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

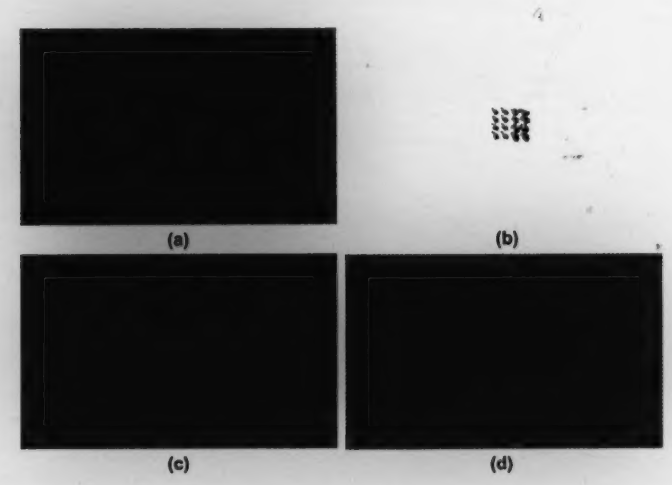


Figure 6.10: The simulation results for the resolution of 80cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

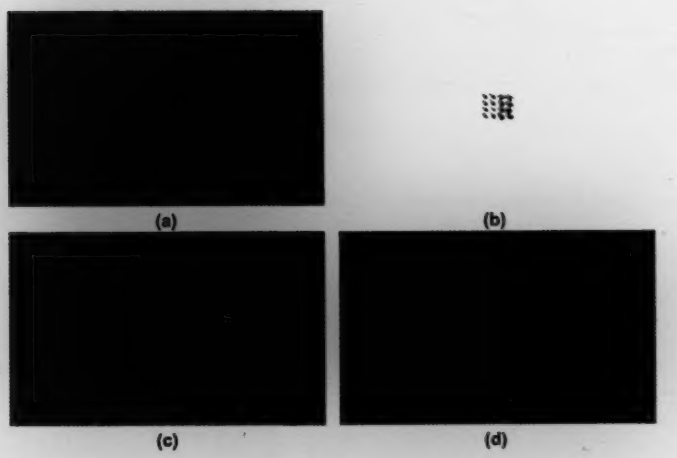


Figure 6.11: The simulation results for the resolution of 100cm per pixel. (a) corresponds to the input left-view image. (b) corresponds to segmented and labelled result, (c) is the disparity map from stereo algorithm. (d) is the predicted 3D scene geometry.

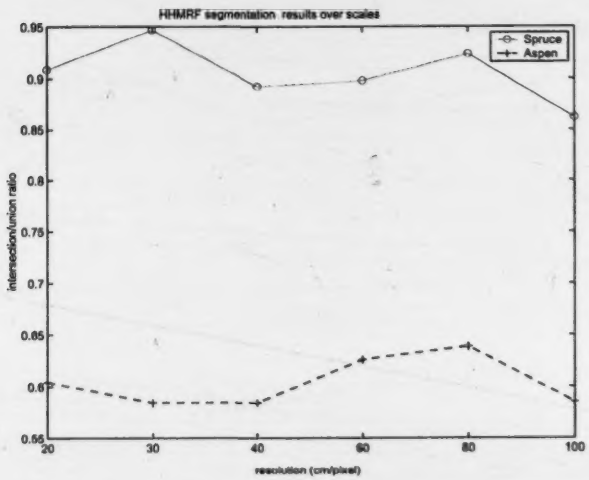


Figure 6.12: This figure shows detection and labelling of trees as a function of image resolution, where the function values equally represent the corresponding confidences. (solid line: Spruce, dashed:Aspen).

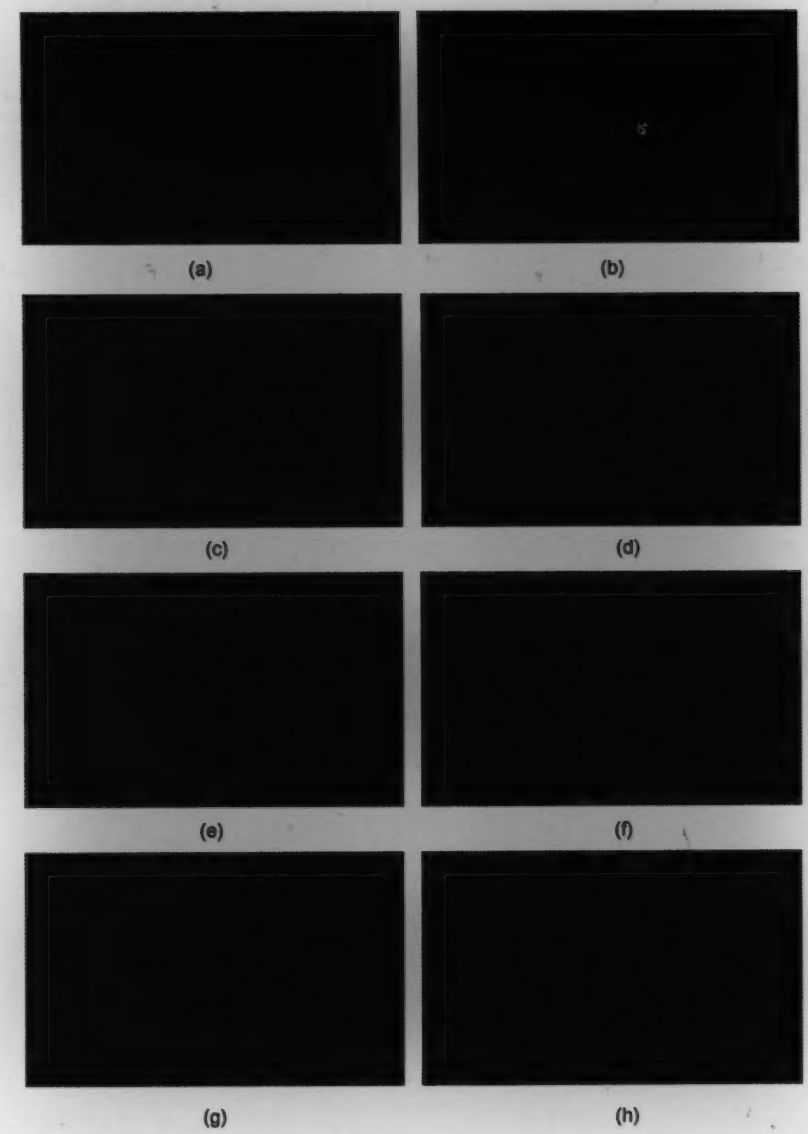


Figure 6.13: This figure shows the robustness of the 3D fitting module when the images are under-segmented. The left view of the input images pair is shown in a), and its segmentation result is shown in b) where several tree regions are merged together due to the overlapping of these trees in the input image. c) and d) show the initial geometric scene when start from $K^{(0)} = 1$ and $K^{(0)} = 2$ per segment, respectively. e) and f) show the corresponding intermediate results at iteration 10. g) and h) show the corresponding final geometric scene results.

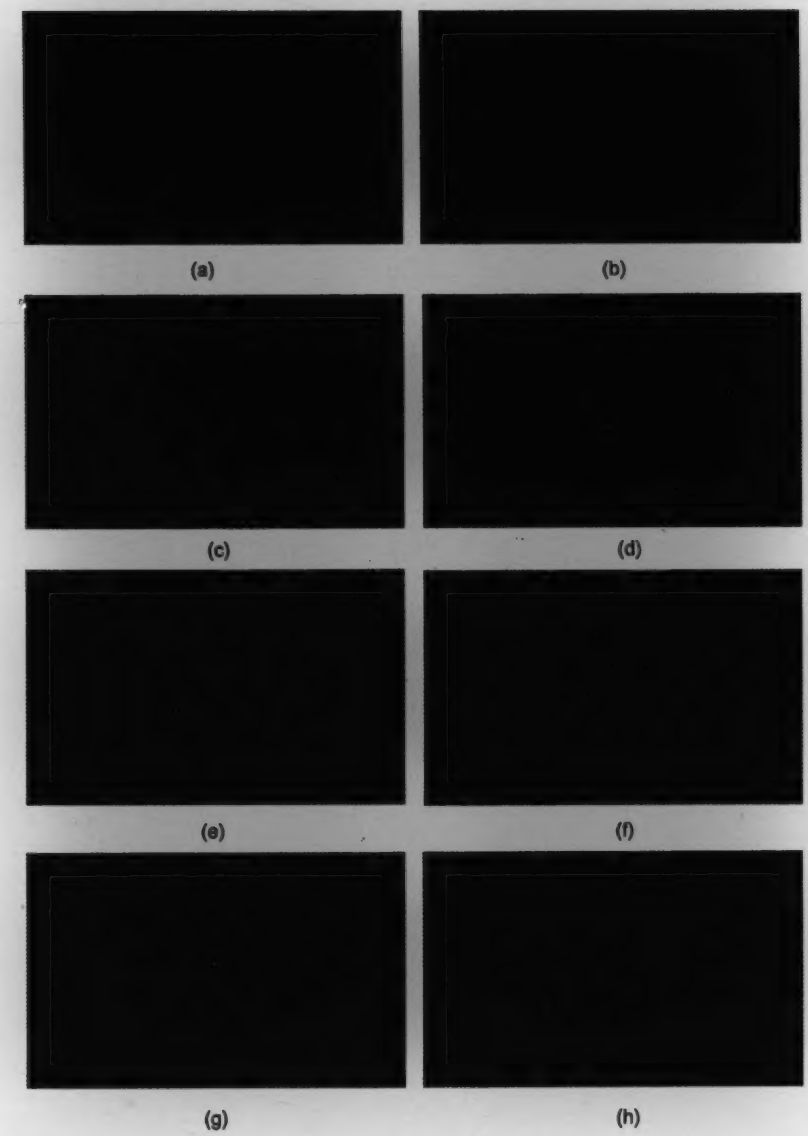


Figure 6.14: This figure shows the corresponding side-view of the 3D fitting dynamic in Fig. 6.13. The left view of the input images pair is shown in a), and its side-view is shown in b). Starting from fitting one tree per segment as shown in c), the side-view of the scene geometry is shown in e); after convergence, the side-view of the scene geometry is shown is g). Similarly, d) shows the initial fitting scene geometry with the side-view shown in f), and h) shows side-view of the final scene geometry, respectively.

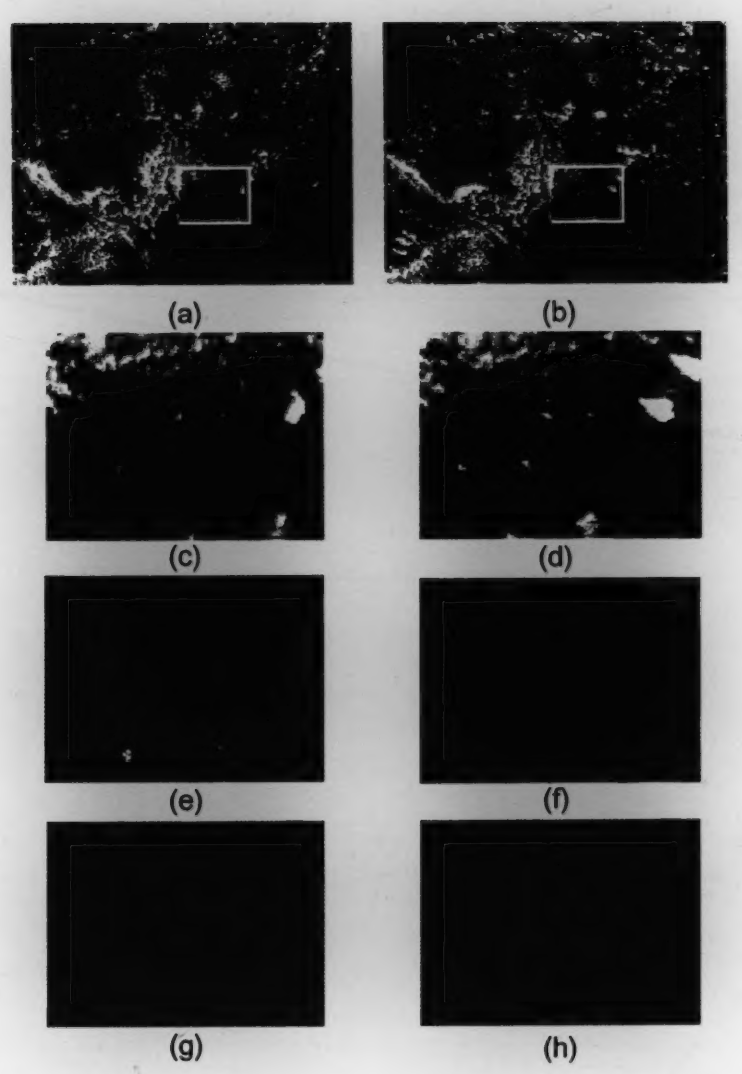


Figure 6.15: This figure shows the experiment of applying the FORIUS system to a real forestry plot from Timberline dataset. a) and b) show the left and right views of the forestry image pair. The white box of a) and b) form the input image pairs, as shown in c) and d), respectively. e) shows the segmentation result of the left view c). f) shows the disparity map of the left view c). The 3D fitting module starts from g) and ends at h).

Chapter 7

Conclusion and Future Work

This PhD thesis is concerned with the entire cycle of Image Understanding (IU): the generation of 3D object hypotheses (inverse modelling) from images and their projections back onto image data (forward modelling), under the Bayesian paradigm. Each subprocess is framed as a local optimization problem based on a component model and related observations. The end result is an IU system that not only provides a symbolic description of scene but also generates fully 3D geometry of the scene being sensed. The system is developed in the context of the need for forestry inventory within the province of Alberta and so most of the examples deal with the interpretation of airborne forestry images.

Controlled simulations with full ground truth were conducted on the synthetic images. Given the success of this, we plan to further investigate the robustness of our proposed framework against broader conditions on the synthesized data set: including

- the study of the forestry image data with various types of mixture and various scales of overlap
- the study of the forestry data with various (textures) seasons
- the study of the forestry data with various lighting conditions.

On the real data side, we conducted experiments on the Timberline airborne image dataset. Due to the wide-basedline stereo nature (Refer to Chapter 6.5 for more description) of the stereo image pair, our stereo module which is designed for standard stereo setting could not work to its full potential. Albeit, our results were quite promising demonstrating a number of principals. Firstly, the stereo and 3D model fitting can be significantly enhanced by using supervised learning and region annotation processes. Secondly, the feedback between image prediction and 3D model extraction converged to interpretations that would not have occurred by using only "open loop" procedures. These interpretations were consistent with ground truth.

Airborne stereo is still a fragile technology. Possible alternatives, or additions, to the prediction of depth for this could be the use the aforementioned taper function approach instead of the stereo module. Due to the rapid progress of the related hardware techniques, we can expect to have higher resolution multi- and hyper-spectral image data available. Further, LIDAR[1] is also emerging as an additional

[1]LIDAR refers to the airborne laser scanner which operates by measuring the return time of

source for measuring the depth information of the dominant trees. This data would complement the high-resolution imagery data, being typically of lower resolution, but with accurate depth measurements. High resolution multi- and even hyper-spectral stereo imagery data, together with LIDAR data and known taper functions, will provide much richer data for the FORIUS system for annotating and fitting 3D tree models.

Credible Image Understanding systems must provide some type of validation or consistency procedures. In this thesis the key idea has been to integrate these into the design of the system itself. In the 3D fitting module of FORIUS, consistency is inherent as part of the error measurement. This results in a 3D forestry scene geometry that is most parsimonious with the hypotheses generated from different sources of information: labelled segmentation, stereo and the expert prior knowledge. The Bayesian formulation has allowed us to formulate such processes in a consistent fashion over all components and combine evidence from different sources and formats.

On the real data side, we conducted experiments on the Timberline airborne image dataset. Due to the wide-basedline stereo nature (Refer to Chapter 6.5 for more description) of the stereo image pair, our stereo module which is designed for standard stereo setting could not work to its full potential. A possible rescue for this is to use the aforementioned taper function approach instead of the stereo module. However, due to the rapid progress of the related hardware techniques, we can expect to have more image data available that have higher resolution, and obey the standard stereo setting in the near future. Besides, the LIDAR [2] comes to be a prominent role in measuring the depth information of the dominant trees. The LIDAR data nicely complements the high-resolution imagery data, since the LADAR data are typically in lower resolution, but with accurate depth measurements. In another word, given both the stereo imagery data and the corresponding LIDAR data, a natural and promising way is to extend the FORIUS system, by incorporating the LADAR data as a new input source in the 3D fitting module.

In the future, to make further evaluation of the system performance, it would be helpful to have images from novel views (of the same scene) as hold out data,

emitted laser pulse and then calculating the range accordingly. LIDAR is also called LADAR for laser radar in literature.

[2]LIDAR refers to the airborne laser scanner which operates by measuring the return time of emitted laser pulse, then calculating the range accordingly. LIDAR is also called LADAR for laser radar in literature.

and have the difference between the hand labelled regions and the system predicted output as the objective criteria for evaluation.

Bibliography

[1] Persistence of vision raytracer: Pov-ray version 3.05. http://www.povray.org/.

[2] D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for boltzmann machine. *Cognitive Science*, 9:147–169, 1985.

[3] N. J. Adams and C. K. I. Williams. Dynamic trees: Learning to model outdoor scenes. In *Proceedings of the European Conference on Computer Vision 2002 (Lecture Notes in Computer Science)*. Springer-Verlag, 2002.

[4] S. Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.

[5] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[6] S.T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3:17–32, 1989.

[7] M. J. Beal, H. Attias, and N. Jojic. Audio-video sensor fusion with probabilistic graphical models. In *European Conference of Computer Vision*, pages 736–752, 2002.

[8] M.J. Beal and Z. Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. In M. J. Bayarri et. al., editor, *Bayesian Statistics 7*. Oxford University Press, 2003.

[9] P.r N. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–262, 1996.

[10] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 23:259–302, 1986.

[11] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–92, 1996.

[12] C. Bouman and B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(2):99–113, 1991.

[13] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, 1994.

[14] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[15] T. Brandtberg. Remote sensing for forestry applications - a historical retrospect. 1999.

[16] T. Brandtberg and F. Walter. Automated delineation of individual tree crowns in high spatial resolution aerial images by multiple-scale analysis. *Machine Vision and Applications*, 11:64–73, 1998.

[17] T. Caelli, L. Cheng, and Q. Fang. A bayesian approach to image understanding: From images to virtual forests (invited talk). In *16th International Conference on Vision Interface*, 2003.

[18] G. Casella and C.P. Robert. Rao-blackewellization of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[19] G. Celeux, D. Chauveau, and J. Diebolt. On stochastic versions of the EM algorithm. Technical Report RR-2514, 1995.

[20] H. Cheng and C. A. Bouman. Multiscale bayesian segmentation using a trainable context model. *IEEE Transactions on Image Processing*, 10(2):460–474, April 2001.

[21] L. Cheng, T. Caelli, and V. Ochoa. A trainable hierarchical hidden markov tree model for color image annotation. In *International Conf. on Pattern Reconition*, 2002.

[22] H. Choi and R. Baraniuk. Multiscale document segmentation using wavelet-domain hidden markov models. In *Proc. of IST/SPIE's 12th Annual Symp. - Electronic Imaging 2000*, 2000.

[23] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on Signal Processing*, 1997.

[24] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Em, mcmc, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50:45–71, 2003.

[25] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[26] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[27] I.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture*. Ellis Horwood Ltd., Wiley and Sons, 1979.

[28] N. De Freitas, P. Hjen-Srensen, M. Jordan, and S. Russell. Variational mcmc. In *Uncertainty in Artificial Intelligence*, pages 120–127, 2001.

[29] B. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *Neural Information Processing Systems Conference*, 2001.

[30] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, London, 1997.

[31] A. Gelman and X. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *IEEE Transactions on Medical Imaging*, 13:163–185, 1998.

[32] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(6):721–741, 1984.

[33] B. Gidas. A renormalization group approach to image processing problems. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(2):164–180, 1989.

[34] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice.* Chapman and Hall, London, 1996.

[35] M. D. Gillis and D. G. Leckie. Forest inventory update in canada. *The Forestry Chronicle*, 72(2):138–156, 1996.

[36] S. J. Godsill. On the relationship between mcmc model uncertainty methods. *Journal of Computational and Graphical Statistics*, 10:230–248, 2001.

[37] F.A. Gougeon. A crown following approach to the automatic delineation of individual tree crowns in high spatial resolution aerial images. *Canadian Journal of Remote Sensing*, 21(3):274–284, 1995.

[38] F.A. Gougeon. Automatic individual tree crown delineation using a valley-following algorithm and rule-based system. In *Proc. Int'l Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 11–23, Victoria, B.C., Canada, 1999.

[39] P. J. Green. Reversible jump mcmc computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[40] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statististical Society B*, 56:549–603, 1994.

[41] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society*, 56(4):549–603, 1994.

[42] R. Hall. *Photogrammetry and Photo Interpretation (Lecture Notes of Forestry Engineering 201).* University of Alberta, 1996.

[43] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[44] D. Higdon, J. Bowsher, V. Johnson, T. Turkington, D. Gilland, and R. Jaszczak. Fully bayesian estimation of gibbs hyperparameters for emission computed tomography data. *IEEE Transactions on Medical Imaging*, 16:516–526, 1997.

[45] S. Huang. Ecologically based individual tree volume estimation for major alberta tree species. Technical report, Land and Forest Sevices, Forest Management Division, Alberta environmental Protection, Edmonton, 1994.

[46] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 5(3):267–287, 1983.

[47] R. M. Gray J. Li. *Image Segmentation and Compression Using Hidden Markov Models.* Kluwer Academic Publishers, 2000.

[48] M. Jordan and C. Bishop. *An Introduction to Graphical Models.* 2004. In Press.

[49] A. Khuri. *Advanced Calculus with Applications in Statistics.* Wiley series in probability and statistics. Wiley-Interscience, 2002.

[50] D. J. King. Airborne remote sensing in forestry: Sensors, analysis and applications. *The Forestry Chronicle, Special Issue on Remote Sensing*, 76(6):25–42, 2000.

[51] I. S. Korpela. 3d matching of tree tops using digitized panchromatic aerial photos. Licentiate thesis, University of Helsinki, Department of Forest Resource Management, Faculty of Agriculture and Forestry, December 2000.

[52] M. Larsen and M. Rudemo. Optimizing templates for finding trees in aerial photographs. *Pattern Recognition Letters*, 19:1153–1162, 1998.

[53] M. A. R. Leisink and H. J. Kappen. A tighter bound for graphical models. *Neural Computation*, 13(9):2149–2171, 2001.

[54] M. Luettgen, W. Karl, A. Willsky, and R. Tenney. Multiscale representations of markov random fields. *IEEE Trans. Signal Process.*, 41(12):3377–3395, 1993.

[55] D. MacKay. Ensemble learning and evidence maximization, 1995.

[56] D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, NATO Science Series, pages 175–204. Kluwer Academic Press, 1998.

[57] D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999.

[58] B.S. Manjunath and R. Chellappa. Unsupervised texture segmentation using random field models. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(5):478–482, May 1991.

[59] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV 2001*, 2001.

[60] G. McLachlan and K. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker Inc., N.Y., 1988.

[61] G. McLachlan and T. Krishnan. *The EM algorithm and Extensions.* Wiley series in probability and statistics. John Wiley and Sons, 1997.

[62] T. Minka. Using lower bounds to approximate integrals. 2001.

[63] R. Neal. *Bayesian Learning for Neural Networks.* Number 118 in Lecture Notes in Statistics. Springer-Verlag, New York, 1996.

[64] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models.* Kluwer, 1998.

[65] R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3):705–767, 2003.

[66] S. Nielsen. The stochastic em algorithm: Estimation and asymptotic results. *Bernoulli*, 6:457–489, 2000.

[67] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

[68] A.J. Pinz. A computer vision system for the recognition of trees in aerial photographs. In *J.C.Tilton (ed.), Multisource Data Integration in Remote Sensing*, pages 111–124, Maryland: NASA, 1991.

[69] R.J. Pollock. The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown image model. *University of British Columbia, Vancouver, Canada*, 1996. PhD Thesis.

[70] D. Pouliot, D. J. King, F. W. Bell, and D. G. Pitt. Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration. *Remote Sensing of Environment*, 82:322–334, 2002.

[71] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: the Art of Scientific Computing.* Cambridge University Press, 1992.

[72] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[73] J. M. Rehg, K. P. Murphy, and P. W. Fieguth. Vision-based speaker detection using bayesian networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 110–116.

[74] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1983.

[75] C. P. Robert. *The Bayesian Choice: from Decision-Theoretic Motivations to Computational Implementation.* Springer-Verlag, New York, second edition, 2001.

[76] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

[77] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IJCV 47(1/2/3):7-42, April-June*, 2002.

[78] L. Shapiro and G. Stochman. *Computer Vision.* Prentice Hall, New Jersey, U.S, 2001.

[79] J. Sun, H. Shum, and N. Zheng. Stereo matching using belief propagation. In *ECCV*, 2002.

[80] R.H. Swendsen and J.S. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Phys. Rev. Lett.*, 58:86–88, 1987.

[81] Z. Tu, X Chen, A.L. Yuille, and S. Zhu. Image parsing: unifying segmentation, detection and recognition. 2003.

[82] Z. Tu and S. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):657–673, 2002.

[83] M. Unser and M. Eden. Multiresolution feature extraction and selection for texture segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11:717–728, 1989.

[84] J. Uuttera, A. Haara, T. Tokola, and M. Maltamo. Determination of spatial distribution of trees from digital aerial photographs. *Forest Ecology and Management*, 110:275–282, 1998.

[85] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Dept. of Statistics, Sep. 2003.

[86] G.C.G. Wei and M. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[87] E. W. Weisstein. Superellipsoid. from mathworld–a wolfram web resource. http://mathworld.wolfram.com/Superellipsoid.html.

[88] D. H. Wolpert and C. E. M. Strauss. What bayes has to say about the evidence procedure. In *Maximum Entropy and Bayesian Methods*, 1993.

[89] C. K. I. Williams X. Feng and S. N. Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 1999. submitted.

[90] E.P. Xing, M.I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence*, 2003.

[91] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *IJCAI*, 20, 2001.

[92] A.L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *Neural Computation*, 15(4):915–936, April 2003.

[93] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1), 2001.

Appendix A

Appendix for Chapter 3.6

A.1 Nomenclature of the segmentation and labelling component

For the image segmentation module, we adopt the following notations:

X , X^l or X^{ALL}	MRF layer for the hidden label sites; l for the l_{th} layer; ALL for all layers
Y, Y^l or Y^{ALL}	random field layer for the observed image sites; l for the l_{th} layer; ALL for all layers
$c \in [1..C]$	label c from the set of labels $[1..C]$
$l \in [1..L]$	l_{th} layer of HHMT model, from the set of layers $[1..L]$
$x_i^l \in X^l$	i_{th} hidden site from layer l
$y_i^l \in Y^l$	i_{th} observed site from layer l
∂l	neighbouring layers of layer l
$A^+_{l-1,l}/A^-_{l,l-1}$	upward transition matrix from layer $l-1$ to layer l / the downward transition matrix from layer l to layer $l-1$
B_l	observation matrix for the l_{th} layer
π_i^l	prior probability of label $i \in [1..C]$ at layer l

Appendix B

Appendix for Chapter 2.5

B.1 The Derivation of Eq. (2.58) of the Stochastic EM

$$
\begin{aligned}
\hat{\sigma}_S^2 &= E\left[\left(E[\phi(x,y)] - \frac{1}{S}\sum_s \phi(x^{(s)},y)\right)^2\right] \\
&= E[\phi(x,y)]^2 - 2E[\phi(x,y)]^2 + \frac{1}{S^2}\sum_{s,s'} E[\phi(x^{(s)},y)\phi(x^{(s')},y)] \\
&= -E[\phi(x,y)]^2 + \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)^2] + \frac{1}{S^2}\sum_{s\neq s'} E[\phi(x^{(s)},y)]E[\phi(x^{(s')},y)] \\
&= -E[\phi(x,y)]^2 + \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)^2] \\
&\quad + \frac{1}{S^2}\sum_{s,s'} E[\phi(x^{(s)},y)]E[\phi(x^{(s')},y)] - \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)]^2 \\
&= -E[\phi(x,y)]^2 + \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)^2] + E[\phi(x,y)]^2 - \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)]^2 \\
&= \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)^2] - \frac{1}{S^2}\sum_s E[\phi(x^{(s)},y)]^2 \\
&= \frac{1}{S}E[(\phi(x,y) - E[\phi(x,y)])^2] \\
&= \frac{1}{S}\sigma^2.
\end{aligned}
$$

B.2 Proof of Eq. (2.16)

We can prove Eq. (2.16) by analyzing the critical point of $\mathcal{L}(\theta, q)$. Since by definition

$$\mathcal{L}(\theta, q) = \inf\{L(\theta; y), \forall\theta\} \tag{B.1}$$

so $\mathcal{L}(\theta, q)$ is a pointwise infimum of $L(\theta; y)$, which establishes $\mathcal{L}(\theta, q)$ as a concave function [14]. Add a Lagrange multiplier to the constraint

$$\int_x q(x)\mathbf{d}x = 1, \tag{B.2}$$

$$\overline{\mathcal{L}}(\theta, q) = \int_x q(x)\log\frac{p(x,y|\theta)}{q(x)}\mathbf{d}x + \lambda(1 - \int_x q(x)\mathbf{d}x).$$

By taking the derivative with respect to $q(x)$, $\frac{\delta\overline{\mathcal{L}}(\theta,q)}{\delta q(x)} = 0$, we have

$$q(x) = p(x, y|\theta)\exp\{-1 - \lambda\}. \tag{B.3}$$

And by taking the derivative with respect to λ, we get back Eq. (B.2) and apply to Eq. (B.3)

$$q(x) = \frac{p(x,y|\theta)}{\int_x p(x,y|\theta)\mathbf{d}x} = p(x|y,\theta). \tag{B.4}$$

Apply Eq. (B.4) back to Eq. (2.15), we see that

$$L(\theta; y) = \mathcal{L}(\theta, q) \quad \textit{when } q(x) = p(x|y,\theta).$$

Therefore, the solution at the critical point in Eq. (B.4) is actually the global maximizer for $\mathcal{L}(\theta, q)$.

B.3 Proof of the Equivalence of Two Lower Bounds

Here we prove the equivalence of the lower bound $\mathcal{L}(\mu, q)$, which is derived from the conventional EM in Eq. (2.14), and the lower bound $\mathcal{L}(\mu, \theta)$, which is derived from the variational EM, and is restricted within the exponentional family.

Let $p(x, y|\theta) = \exp\{< \theta, \phi(x, y) > -Z(\theta)\}$, we have

$$\begin{aligned}
\mathcal{L}(\mu, q) &= \int_x q(x) \log \frac{p(x, y|\theta)}{q(x)} \mathbf{d}x \\
&= \int_x q(x) \log p(x, y|\theta) \mathbf{d}x - \int_x q(x) \log q(x) \mathbf{d}x \\
&= \int_x q(x) < \theta, \phi(x, y) > \mathbf{d}x - \int_x q(x) \log q(x) \mathbf{d}x - Z(\theta) \\
&= < \theta, E_q[\phi(x, y)] > - \int_x q(x) \log q(x) \mathbf{d}x - Z(\theta).
\end{aligned}$$

If we let $\mu = E_q[\phi(x, y)]$, the expectation of sufficient statistics, by Eq. (2.30) that $\mathcal{L}(\mu, \theta) =< \theta, \mu > -Z_y^*(\mu) - Z(\theta)$, we prove that $\mathcal{L}(\mu, q) = \mathcal{L}(\mu, \theta)$.

B.4 Derivations that Related to Variational Bayes (VB)

Given the joint pdf is already defined, and define

$$Z_y(\eta) = \log \int_\theta \exp\{< \eta, \psi(\theta) > -Z(\theta)\} \mathbf{d}\theta.$$

We now write down the marginal posterior probabilities for the quantities of interest. First, the joint posterior probability over (x, θ) is

$$\begin{aligned}
p(x, \theta|y, M) &= \frac{p(x, y, \theta|M)}{\int_\theta \int_x p(x, y, \theta|M) \mathbf{d}x \mathbf{d}\theta} \\
&= \exp\{< \theta, \phi(x, y) > + < \eta, \psi(\theta) > -Z(\theta) - Z_y(\eta)\}.
\end{aligned}$$

Second, the marginal of θ given observed data and model is

$$\begin{aligned}
p(\theta|y, M) &= \int_x p(x, \theta|y, M) \mathbf{d}x \\
&= \exp\{< \eta, \psi(\theta) > -Z(\theta) - Z_y(\theta) - Z_y(\eta)\}.
\end{aligned}$$

Third, the marginal of x given observed data and model is

$$\begin{aligned}
p(x|y, M) &= \int_\theta p(x, \theta|y, M) \mathbf{d}\theta \\
&= \exp\{Z_y(x, \eta) - Z_y(\eta)\}
\end{aligned}$$

with

$$Z_y(x, \eta) = \log \int_\theta \exp\{< \theta, \phi(x, y) > + < \eta, \psi(\theta) > -Z(\theta)\} \mathbf{d}\theta.$$

It is obvious to establish the equality of $\mathcal{L}(\eta, \mu_\psi)$ and $\mathcal{L}(\eta, q(x))$, as

$$\begin{aligned}
\mathcal{L}(\eta, \mu_\psi) &= < \eta, \mu_\psi > -Z_y^*(\mu_\psi) - Z(\eta) \\
&= < \eta, \int_\theta q(\theta) \psi(\theta) \mathbf{d}\theta > - \int_\theta q(\theta) (\log q(\theta) + Z(\theta) - Z_y(\theta)) \mathbf{d}\theta - \int_\theta q(\theta) Z(\eta) \mathbf{d}\theta \\
&= \int_\theta q(\theta) \log \frac{\exp\{\eta, \psi(\theta) - Z(\eta) + Z_y(\theta) - Z(\theta)\}}{q(\theta)} \mathbf{d}\theta \\
&= \mathcal{L}(\eta, q(x)).
\end{aligned}$$

Similarly, we have the equality of $\mathcal{L}((\theta,\eta),(\mu_\phi,\mu_\psi))$ and $\mathcal{L}((\theta,\eta),q(x)q(\theta))$, as

$$\begin{aligned}
\mathcal{L}((\theta,\eta),(\mu_\phi,\mu_\psi)) &= <\eta,\mu_\psi> - Z_y^*(\mu_\phi,\mu_\psi) - Z(\eta) \\
&= \int_\theta q(\theta)(<\eta,\psi(\theta)> - Z(\eta) - \log q(\theta) - Z(\theta) \\
&+ <\theta, \int_x q(x)\phi(x,y)\mathbf{d}x> - \int_x q(x)\log q(x)\mathbf{d}x)\mathbf{d}\theta \\
&= \int_\theta q(\theta)\left(\log\frac{\exp\{<\eta,\psi(\theta)> - Z(\eta) - Z(\theta)\}}{q(\theta)} + \int_x q(x)\log\frac{\exp\{<\theta,\phi(x,y)>\}}{q(x)}\mathbf{d}x\right) \\
&= \int_\theta q(\theta)\int_x q(x)\log\frac{\exp\{<\eta,\psi(\theta)> + <\theta,\phi(x,y)> - Z(\eta) - Z(\theta)\}}{q(x)q(\theta)}\mathbf{d}x\mathbf{d}\theta \\
&= \mathcal{L}((\theta,\eta),q(x)q(\theta)).
\end{aligned}$$

We can then verify that $\mathcal{L}((\theta,\eta),q(x)q(\theta))$ is indeed the lower bound of $L(M;y)$ by

$$\begin{aligned}
L(M;y) - \mathcal{L}((\theta,\eta),q(x)q(\theta)) &= \int_\theta\int_x \log\frac{p(y|M)q(x)q(\theta)}{p(x,y,\theta|M)}\mathbf{d}x\mathbf{d}\theta \\
&= KL(q(x)q(\theta)||p(x,\theta|y,M)).
\end{aligned}$$

It is not hard to show that the log-evidence can be decomposed into two functions

$$\begin{aligned}
L(M;y) &= \log\int_\theta\int_x \exp\{<\theta,\phi(x,y)> + <\eta,\psi(\theta)> - Z(\theta) - Z(\eta)\}\mathbf{d}x\mathbf{d}\theta \\
&= \log\int_\theta \exp\{<\eta,\psi(\theta)> - Z_y(\theta) - Z(\theta)\}\mathbf{d}\theta - Z(\eta) \\
&= Z_y(\eta) - Z(\eta).
\end{aligned}$$

We already know that $Z(\eta)$ is convex, and it is easy to show $Z_y(\eta)$ is also a convex function, by

(a) Taking the first order derivative of $Z_y(\eta)$ with respect to η, and getting

$$\mu_\psi = \nabla Z_y(\eta)$$

(b) Taking the second order derivative, and having

$$\nabla\nabla Z_y(\eta) = E[\psi(\theta)^2] - E^2[\psi(\theta)] \geq 0.$$

Recall when at optimality $q(\theta) = p(\theta|y,M)$. Thus the Fenchel conjugate is

$$\mu_\psi = \int_\theta p(\theta|y,M)\psi(\theta)\mathbf{d}\theta = \int_\theta q(\theta)\log\frac{\int_x p(x,y|\theta)\mathbf{d}x}{q(\theta)}\mathbf{d}\theta. \tag{B.5}$$

Substitute it into Eq. (2.47) and we have

$$L(M;y) \geq <\eta,\mu_\psi> - Z_y^*(\mu_\psi) - Z(\eta) = \mathcal{L}(\eta,\mu_\psi). \tag{B.6}$$

This naturally leads to the EM-type algorithm. Notice $Z_y^*(\mu_\psi)$ is not a convex function, as is clear from Eq. (2.48). We now further lower-bound it. Since from variation EM, we have

$$L(\theta;y) = \log p(y|\theta) = Z_y(\theta) - Z(\theta). \tag{B.7}$$

The Fenchel duals of $Z_y(\theta)$ are

$$\begin{aligned} Z_y(\theta) &= \sup_{\mu_\phi}\{< \theta, \mu_\phi > - Z_y^*(\mu_\phi)\} \\ Z_y^*(\mu_\phi) &= \sum_{\theta}\{< \mu_\phi, \theta > - Z_y(\theta)\}, \end{aligned}$$

and we know the log-partition function is convex, as

$$Z_y^*(\mu_\phi) = \int_x q(x) \log q(x) \mathbf{d}x. \tag{B.8}$$

Substitute Eq. (B.8) into Eq. (B.7)

$$L(\theta; y) \geq < \theta, \mu_\phi > - Z_y^*(\mu_\phi) - Z(\theta). \tag{B.9}$$

Then substitute Eq. (B.9) into Eq. (2.48), we find Eq. (2.49).

Now, let us look at the EM-type algorithm in Eq. (2.51) and Eq. (2.52), and compute the update equations by finding the critical point. First, instead of computing u_ψ^* directly, we compute $q(x)$ by taking the derivative of the lower bound wrt $q(x)$ and equate to zero, as

$$0 = \frac{\delta \mathcal{L}}{\delta q(x)} = < \mu_\theta, \phi(x, y) > - \log x - 1.$$

With the constraint that $\int_x q(x)\mathbf{d}x = 1$, we have

$$q(x) = \frac{\exp\{< \mu_\theta, \phi(x, y) >\}}{\int_x \exp\{< \mu_\theta, \phi(x, y) >\}\mathbf{d}x}$$

which, at the optimal point, leads to Eq. (2.53). Second, by

$$0 = \frac{\delta \mathcal{L}}{\delta \theta}$$

we recover $\mu_\phi = \nabla Z(\theta)$. Third, by

$$0 = \frac{\delta \mathcal{L}}{\delta q(\theta)} = < \eta, \psi(\theta) > + < \theta, \mu_\phi > - \log q(\theta) - Z(\theta) - 1$$

and by the probability constraint $\int_\theta q(\theta)\mathbf{d}\theta = 1$, we have

$$q(\theta) = \frac{\exp\{< \eta, \psi(\theta) > + < \theta, \mu_\phi > - Z(\theta)\}}{\int_\theta \exp\{< \eta, \psi(\theta) > + < \theta, \mu_\phi > - Z(\theta)\}\mathbf{d}\theta}.$$

Similarly, it leads to Eq. (2.54) at the optimality. Fourth, by

$$0 = \frac{\delta \mathcal{L}}{\delta \eta}$$

we recover the fix point equation $\mu_\psi = \nabla Z(\eta)$.

B.5 Proof of the M–H algorithm

We sketch the proof that by construction the Metroplis-Hastings algorithm fullfil the detailed balance condition. Note the proof here is only a sketch, and the reader could consult [43] for a complete proof. The only reason to include this proof here is to maintain the integrity of the style we adopt in reviewing the related statistical literature.

First, consider the case when the proposed sample x^* is accepted, $x^{(s+1)} = x^*$. The transition kernel is then

$$K(x^{(s+1)}|x^{(s)}) = q(x^{(s+1)}|x^{(s)})A(x^{(s)}, x^{(s+1)}).$$

Start from the LHS of the detailed balance condition, we can prove the equality to the RHS holds

$$\begin{aligned}
K(x^{(s+1)}|x^{(s)})p(x^{(s)}) &= q(x^{(s+1)}|x^{(s)})A(x^{(s)}, x^{(s+1)})p(x^{(s)}) \\
&= q(x^{(s+1)}|x^{(s)}) \min\left\{1, \frac{q(x^{(s)}|x^*)p(x^*)}{q(x^*|x^{(s)})p(x^{(s)})}\right\} p(x^{(s)}) \\
&= \min\left\{q(x^{(s+1)}|x^{(s)})p(x^{(s)}), q(x^{(s)}|x^*)p(x^*)\right\} \\
&= \min\left\{q(x^{(s)}|x^{(s+1)})p(x^{(s+1)}), q(x^{(s+1)}|x^{(s)})p(x^{(s)})\right\} \\
&= K(x^{(s)})p(x^{(s)}|x^{(s+1)}).
\end{aligned}$$

Second, consider the remaining case when the proposed sample x^* is rejected, thus $x^{(s+1)} = x^{(s)} \neq x^*$. The transition kernel is

$$K(x^{(s+1)}|x^{(s)}) = \int_{x^*} q(x^*|x^{(s)})(1 - A(x^{(s)}, x^*))\mathbf{d}x^*.$$

Then again, from the LHS, we can prove the detailed balance condition still holds

$$\begin{aligned}
K(x^{(s)}|x^{(s)})p(x^{(s)}) &= \int_{x^*} q(x^*|x^{(s)}) \left(1 - \min\left\{1, \frac{q(x^{(s)}|x^*)p(x^*)}{q(x^*|x^{(s)})p(x^{(s)})}\right\}\right) \mathbf{d}x^* p(x^{(s)}) \\
&= \int_{x^*} q(x^*|x^{(s)})p(x^{(s)}) \left(\max\left\{0, \frac{q(x^*|x^{(s)})p(x^{(s)}) - q(x^{(s)}|x^*)p(x^*)}{q(x^*|x^{(s)})p(x^{(s)})}\right\}\right) \mathbf{d}x^* \\
&= \int_{x^*} \max\left\{0, q(x^*|x^{(s)})p(x^{(s)}) - q(x^{(s)}|x^*)p(x^*)\right\} \mathbf{d}x^* \\
&= \max\{0, p(x^{(s)}) - \int_x q(x^{(s)}|x^*)p(x^*)\mathbf{d}x^*\} \\
&= p(x^{(s)})K(x^{(s)}|x^{(s)}).
\end{aligned}$$

B.6 Definition of Markov Random Field (MRF) and the Hammersley-Clifford Theorem

Consider undirected graph $G = (V, E)$ as formulated in Sec. 2.4.1. Define $N(i)$ the neighbourhood of node i, for any $i \in V$. Let $X = x_i, i \in V$ be the random field. We say X obeys Markov Random Field (MRF), if and only if

- $p(X) > 0, \forall X$

- $p(x_i|X_{\setminus i}) = p(x_i|x_{N(i)}), \forall i \in V$

where $\cdot\setminus i$ refers to the set of all nodes except node i.

Alternatively, we can view G as partitioned by a set of cliques (fully connected subgraph) $G = \{c, c \in \mathcal{C}\}$. Define $V_c(X_c) = -\theta_c\phi(X_c)$ the potential function for clique c, and $U(x) = \sum_{c\in\mathcal{C}} V_c(X_c)$ the energy function. The Hammersley-Clifford theorem restricts the distributions of MRF model to be within the exponential family (Gibbs distribution), as:

- Random field X is a MRF, $p(X) > 0$, $\forall X$, if and only if

$$p(X) = \frac{1}{Z}\exp\{-\sum_{c\in\mathcal{C}} V_c(X_c)\}$$

with $Z = \int \exp\{-\sum_{c\in\mathcal{C}}\}\nu(dX)$.

B.7 Derivation of the Log-partition Functions

Here we prove that inequality occurred during the lower bound derivation of the log incomplete data function.

The empirical conditional probability function in Eqn.2.66 could be further written as:

$$\begin{aligned} p(\hat{Y}|X,\beta) &= \exp\{\int_x \left(\sum_i \delta(x_i, x) < \beta_x, \phi(x, \hat{y}_i) > -|V_x|Z(\beta_x)\right)\nu(dx)\} \\ &= \exp\{<\beta, \sum_i \phi(x_i, \hat{y}_i) > -Z(\beta, X)\} \end{aligned}$$

where $\delta(\cdot)$ is the Kronecker product and $|V_x|$ is the number of nodes that share the same realization of x in current configuration of X. To simplify the notation, we define the function which partitions essentially the x-dependant partition function $Z(\beta_x)$ of all nodes into disjoint groups, given current configuration of X, as

$$\begin{aligned} Z(\beta, X) &= \int_x |V_x|Z(\beta_x)\nu(dx) \\ &= \sum_i Z(\beta_{x_i}), \end{aligned}$$

and define $< \beta, \sum_i \phi(x_i, \hat{y}_i) >= \int_x \sum_i \delta(x_i, x) < \beta_x, \phi(x, \hat{y}_i) > \nu(dx)$.

As follows, we can write down other probability functions as follows:

$$\begin{aligned} p(\hat{Y}|\theta) &= \int_X p(X, \hat{Y}|\theta)\nu(dX) \\ &= \int_X p(X|\alpha)p(\hat{Y}|X,\beta)\nu(dX) \\ &= \int_X \exp\{<\alpha, \sum_e \phi(x_e) > -Z(\alpha) + <\beta, \sum_i \phi(x_i, \hat{y}_i) > -Z(\beta, X)\}\nu(dX) \\ &= \exp\{Z(\theta) - Z(\alpha)\} \end{aligned}$$

where the log partition function is, and by Jensen's Inequality, could be lower-bounded as:

$$Z(\theta) = \log \int_X \exp\left\{ < \alpha, \sum_e \phi(x_e) > + < \beta, \sum_i \phi(x_i, \hat{y}_i) > - Z(\beta, X) \right\} \nu(dX) \quad \text{(B.10)}$$

$$\geq \int_X q(X) \left\{ < \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \begin{pmatrix} \sum_e \phi(x_e) \\ \sum_i \phi(x_i, \hat{y}_i) \end{pmatrix} > - Z(\beta, X) \right\} \nu(dX) \quad \text{(B.11)}$$

$$- \int_X q(X) \log q(X) \nu(dX) \quad \text{(B.12)}$$

$$= < \theta, \mu(q) > - Z_q(\beta) - Z^*(\mu(q)) \quad \text{(B.13)}$$

where $\mu(q) = E_{q(X)} \begin{bmatrix} \sum_e \phi(x_e) \\ \sum_i \phi(x_i, \hat{y}_i) \end{bmatrix}$. We define the expected partition function $Z_q(\beta) = \int_X q(X) Z(\beta, X) \nu(dX)$, and $Z^*(\mu(q)) = -H(q)$, the Fenchel convex dual of $Z(\theta)$ [14]. $H(\cdot)$ is the entropy (or differential entropy in the continuous case) function.

The goal of the EM algorithm is to maximize the log-incomplete-data function $L(\theta)$, as:

$$\begin{aligned} L(\theta) &= \log p(\hat{Y}|\theta) && \text{(B.14)} \\ &= KL(\mu(q)||\theta) + \mathcal{L}(\mu(q), \theta) && \text{(B.15)} \\ &= Z(\theta) - Z(\alpha) && \text{(B.16)} \\ &\geq < \theta, \mu(q) > - Z_q(\beta) - Z^*(\mu(q)) - Z(\alpha) && \text{(B.17)} \\ &= \mathcal{L}(\mu(q), \theta) && \text{(B.18)} \\ &= -F(\theta, \mu(q)) && \text{(B.19)} \end{aligned}$$

where $\mathcal{L}(\mu(q), \theta)$ is the lower bound of the objective function. The equality holds when (a) $L(\theta)$ is a convex function; and (b) we reach the optimality of the solution space of $q(X)$, that is, when we get the exact solution as

$$q(X) = p(X|\hat{Y}, \theta). \quad \text{(B.20)}$$

However, $L(\theta)$ is usually not a convex function, as evidenced by Eq. (B.14) where it equals the subtraction of two convex functions, this is the same situation as dealt with in the concave-convex (CCCP) procedure [92]. In fact, the first-order variational EM formulation we employed here is the same as CCCP.

B.8 Derivation of the Updating Equations in Exact–EM

In this section, with the HMRF model, we consider the EM procedure with exact inference in E step[1]. From the previous sections, we prove that maximizing the incomplete data likelihood is equivalent to the maximization of the lower bound:

$$\mathcal{L}(\mu(q), \theta) = < \theta, \mu(q) > - Z_q(\beta) - Z^*(\mu(q)) - Z(\alpha).$$

It is obvious that the lower bounds could be rearranged in a way that we have a distinct lower bound for different scenario, while preserving the optimality.

[1]This procedure is referred to as Exact-EM.

- In the E step, the lower bound is

$$\begin{aligned}\mathcal{L}^E(\mu(q)) =& < \mu(q), \theta > -Z_q(\beta) - Z^*(\mu(q)) \\ = & \int_X q(X) \left(< \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta) \right) \nu(dX) \\ & - \int_X q(X) \log q(X) \nu(dX)\end{aligned} \tag{B.21}$$

with $\mathcal{L}(\mu(q), \theta) = \mathcal{L}^E(\mu(q)) - Z(\alpha)$.

- In the M step

$$\begin{aligned}\mathcal{L}^M(\theta) =& < \theta, \mu(q) > -Z_q(\beta) - Z(\alpha) \\ = & \int_X q(X) \left(< \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta) \right) \nu(dX) \\ & - \log \int_X \exp\{ < \alpha, \sum_e \phi(x_e) > \} \nu(dX)\end{aligned} \tag{B.22}$$

with $\mathcal{L}(\mu(q), \theta) = \mathcal{L}^M(\theta) - Z^*(\mu(q))$. In specific,

  - for the parameter vector α, the lower bound becomes

$$\mathcal{L}^M(\alpha) =< \theta, \mu(q) > -Z(\alpha) \tag{B.23}$$

  - for the parameter vector β,

$$\mathcal{L}^M(\beta) =< \theta, \mu(q) > -Z_q(\beta). \tag{B.24}$$

Firstly, we derive the fix-point equation for updating $\mu(q)$ in the E step. From Eq. (2.72), augmented with the Lagrange multiplier λ_q to enforce the fact that the probability should sum to 1, we redefine the lower bound objective function in the E step as

$$\begin{aligned}\mathcal{L}^E(\mu(q), \lambda_q) &= < \mu(q(X)), \theta^{(t)} > -Z_q(\beta) - Z^*(\mu(q(X))) + \lambda_q(\int_X q(X) \nu(dX) - 1) \\ &= \int_X q(X) (< \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i})) \nu(dX) \\ &\quad - \int_X q(X) \log q(X) \nu(dX) + \lambda_q(\int_X q(X) \nu(dX) - 1).\end{aligned}$$

Obviously this function is a concave function with respect to μ (and also with respect to $q(X)$ since by definition, $\mu(q(X))$ is a linear function of $q(X)$). Instead of directly computing μ, it is much easier to compute $q(X)$ by taking the derivative and set to zero, as

$$0 = \frac{\delta \mathcal{L}^E(\mu(q), \lambda_q)}{\partial q(X)} =< \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i}) - \log q(X) - 1 + \lambda_q. \tag{B.25}$$

Thus, we have the update equation for $q(X)$

$$q(X) = \frac{\exp\{ < \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i}) + \lambda_q \}}{e}. \tag{B.26}$$

To solve the Lagrange multiplier as

$$0 = \frac{\delta \mathcal{L}^E(\mu(q), \lambda_q)}{\partial \lambda_q} = \int_X q(X)\nu(dX) - 1,$$

we get back the constraint $\int_X q(X)\nu(dX) = 1$ and apply to Eq. (B.26), we have

$$e = \int_X \exp\left\{ < \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i}) \right\} \nu(dX) = \exp\{Z(\theta)\}.$$

Substituting this back to Eq. (B.26), we have the iterative update equation

$$q(X) = \exp\left\{ < \alpha, \sum_e \phi(x_e) > + \sum_i \log p(\hat{y}_i | x_i, \beta_{x_i}) - Z(\theta) \right\}. \tag{B.27}$$

$\mu(q)$ can then be computed accordingly.

Secondly, we update the parameters in the M step. To update α, we compute

$$\begin{aligned} 0 = \frac{\delta \mathcal{L}^M(\alpha)}{\delta \alpha} &= \frac{\delta}{\delta \alpha}\{ < \theta, \mu > - Z(\alpha)\} \\ &= E_{q(X)}[\sum_e \phi(x_e))] - E_{p(X|\alpha)}[\sum_e \phi(x_e)]. \end{aligned}$$

Obviously the closed form solution for α does not exist. In general, α can be updated either by the gradient descent type method or by iterative scaling method. In gradient descent method, the search direction is:

$$\frac{\delta \mathcal{L}^M(\alpha)}{\delta \alpha} = E_{p(X|\alpha)}[\sum_c \phi(x_c)] - E_q(X)[\sum_e \phi(x_e)]. \tag{B.28}$$

However, in the one-dimensional case, we can solve α by finding the root of the equation

$$E_q(X)[\sum_e \phi(x_e)] = E_{p(X|\alpha)}[\sum_c \phi(x_c)]. \tag{B.29}$$

To update β, we compute by taking the derivative

$$0 = \frac{\delta \mathcal{L}^M(\beta)}{\delta \beta} = E_{q(X)}[\sum_i \phi(\hat{y}_i|_{x_i=x}))] - E_{q(X)}[|V_x| \nabla Z(\beta_x)].$$

Obviously $\nabla Z(\beta_x) = E_{p(y|x,\beta_x)}[\phi(x,y)]$, so $\beta = \{\beta_x\}$ could be updated as

$$E_{q(X)}[\sum_i \phi(\hat{y}_i|_{x_i=x}))] = E_{q(X)}\left[|V_x| E_{p(y|x,\beta_x)}[\phi(x,y)]\right]. \tag{B.30}$$

In our example, the lower bounds are

- E step

$$\mathcal{L}^E(\mu(q)) = \sum_X q(X) \left(\alpha \sum_{(i,j)\in E} x_i x_j + \sum_i \log p(\hat{y}_i | x_i, \beta) \right) - \sum_X q(X) \log q(X).$$

- M step
  - for the parameter vector α, the lower bound becomes

$$\mathcal{L}^M(\alpha) = \sum_X q(X)\alpha \sum_{(i,j)\in E} x_i x_j - Z(\alpha) \tag{B.31}$$

  - for the parameter vector β,

$$\mathcal{L}^M(\beta, x) = \beta_1 \sum_X q(X) \sum_i (\hat{y}_i|_{x_i=x}) - \frac{\beta_2}{2} \sum_X q(X) \sum_i (\hat{y}_i^2|_{x_i=x}) - \sum_X q(X)|V_x| Z(\beta_x).$$

Then we solve β_x by again taking the derivatives of the lower bound $\mathcal{L}^M(\beta, x)$ and equate to zero, for β_{x1} and β_{x2} respectively

$$0 = \frac{\delta \mathcal{L}^M(\beta, x)}{\delta \beta_{x1}} = \sum_X q(X) \sum_i (\hat{y}_i|_{x_i=x}) - \sum_X q(X)|V_x| \frac{\delta Z(\beta_x)}{\delta \beta_{x1}}. \tag{B.32}$$

$$0 = \frac{\delta \mathcal{L}^M(\beta, x)}{\delta \beta_{x2}} = -\frac{1}{2} \sum_X q(X) \sum_i (\hat{y}_i^2|_{x_i=x}) - \sum_X q(X)|V_x| \frac{\delta Z(\beta_x)}{\delta \beta_{x2}}. \tag{B.33}$$

For the Gaussian case in Eq. (2.70)

$$\frac{\delta Z(\beta_x)}{\delta \beta_{x1}} = \frac{\beta_{x1}}{\beta_{x2}}, \tag{B.34}$$

$$\frac{\delta Z(\beta_x)}{\delta \beta_{x2}} = -\frac{\beta_{x1}^2}{2\beta_{x2}^2} - \frac{1}{2\beta_{x2}}, \tag{B.35}$$

substituting Eq. (B.34, B.35) into Eq. (B.32, B.33), We have, the equations for β_x

$$\sum_X q(X) \sum_i (\hat{y}_i|_{x_i=x}) = \sum_X q(X)|V_x| \frac{\beta_{x1}}{\beta_{x2}} \tag{B.36}$$

$$\sum_X q(X) \sum_i (\hat{y}_i^2|_{x_i=x}) = \sum_X q(X)|V_x| \left(\frac{\beta_{x1}^2}{\beta_{x2}^2} + \frac{1}{\beta_{x2}} \right) \tag{B.37}$$

which could also be formulated as Eq. (2.78, 2.79).

B.9 Derivation of the Updating Equations in NMF–EM

Now we consider the EM procedure with NMF approximation in E step which we call the NMF–EM procedure. Starting from the factorization in Eq. (2.84), we can derive $Z_{\mathcal{H}_0}(\theta)$, based on Eq. (B.10), as[2]

$$\begin{aligned} Z_{\mathcal{H}_0}(\theta) &= \sum_e \log \int_{x_e} \exp\{< \alpha, \phi(x_e) >\} \nu(dx_e) + \sum_i \int_{x_i} p(\hat{y}_i|x_i, \beta) \nu(dx_i) \\ &\geq \sum_e \int_{x_e} \prod_{i \in e} < \alpha, \phi(x_e) > \nu(dx_e) + \sum_i \int_{x_i} q(x_i) \log p(\hat{y}_i|x_i, \beta) \nu(dx_i) - \sum_i Z_i^*(\mu) \\ &= \mathcal{L}_{\mathcal{H}_0}^E(\mu) \end{aligned}$$

[2]To keep the notation simple, sometimes we choose not to append the $\mathcal{H}_0$ identifier, to avoid confusion.

where

$$p(\hat{y}_i|x_i,\beta) = \exp\{<\beta, \sum_i \phi(x_i,\hat{y}_i)> - Z(\beta,x_i)\}$$

$$Z_i^*(\mu) = \log \int_{x_i} q(x_i)\log q(x_i)\nu(dx_i).$$

To keep terms simple, let's define

$$E_{q(x_e)}[<\alpha,\phi(x_e)>] = \sum_e \int_{x_e} \prod_{i\in e} q(x_i) < \alpha,\phi(x_e) > \nu(dx_e)$$

$$E_{q(x_i)}[\log p(\hat{y}_i|x_i,\beta)] = \sum_e \int_{x_e} q(x_i)\log p(\hat{y}_i|x_i,\beta)\nu(dx_i)$$

$$Z_{\mathcal{H}_0}^*(\mu) = \sum_i Z_i^*(\mu).$$

By ignoring the overcounting of edges e, $Z_{\mathcal{H}_0}(\theta)$ could be further simplified as

$$Z_{\mathcal{H}_0}(\theta) = \sum_i Z_i(\theta) \geq \sum_i (<\theta,\mu(q(x_i))> - Z_{q(x_i)}(\beta) - Z_i^*(\mu(q)))$$

where

$$Z_i(\theta) = \int_{x_i} \exp\left\{\alpha x_i \sum_{j\in\partial i}\mu_i + \log p(\hat{y}_i|x_i,\beta)\right\}\nu(dx_i).$$

This leads to the log incomplete data, based on Eq. (B.14), as

$$\begin{aligned} L_{\mathcal{H}_0}(\theta) &= Z_{\mathcal{H}_0}(\theta) - Z_{\mathcal{H}_0}(\alpha) && \text{(B.38)}\\ &\geq \sum_i (<\theta,\mu(q(x_i))> - Z_{q(x_i)}(\beta) - Z_i^*(\mu(q)) - Z_i(\alpha)) && \text{(B.39)}\end{aligned}$$

In the E step, the lower bound, based on Eq. (B.21), is

$$\mathcal{L}_{\mathcal{H}_0}^E(\mu) = E_{q(x_e)}[<\alpha,\phi(x_e)>] + E_{q(x_i)}[\log p(\hat{y}_i|x_i,\beta)] - Z_{\mathcal{H}_0}^*(\mu). \qquad \text{(B.40)}$$

Since this is a concave function, in line with what we did in the Exact–EM section, and from Eq. (B.38, B.40), we come to the update equation in Eq. (2.85).

In the M step, the lower bound, based on Eq. (B.22), reads

$$\mathcal{L}_{\mathcal{H}_0}^M(\theta) = E_{q(x_e)}[<\alpha,\phi(x_e)>] + E_{q(x_i)}[\log p(\hat{y}_i|x_i,\beta)] - Z_{\mathcal{H}_0}(\alpha)$$

where $Z_{\mathcal{H}_0}(\alpha) = \sum_e \int_{x_e} \exp\{<\alpha,\phi(x_e)>\}\nu(dx_e)$. From it we can easily derive Eq. (2.86) for updating α, Eq. (2.87, 2.88) for updating β.

Appendix C

Appendix for Chapter 4.4

C.1 Full Conditional Probability

From Fig. 4.1, we define the node (variable) set $V = \{v\}$, and define $pa[v]$ as the parent nodes of v, $ch[v]$ as the children nodes of v and $\backslash v$ as all nodes except v, and:

$$p(V) = \prod_{v \in V} p(v|pa[v]).$$

That is, the full conditional probability for each node is conditioned on the Markov Blanket [48] defined, in this case, as

$$p(v|V_{\backslash v}) \propto p(v, V_{\backslash v}) \tag{C.1}$$

$$= p(v|pa[v]) \prod_{w \in ch[v]} p(w|pa[w]). \tag{C.2}$$

C.2 Updating σ_y

A conjugate prior [1] is chosen to define the prior $p(\sigma_y^2)$ which is the scaled inverse chi-square [30],

$$\sigma_y^2 \sim Inv - \chi^2(\sigma_{y0}, v_0)$$

where

$$p(\sigma_y^2) = (\frac{\sigma_{y0}^2}{\sigma_y^2})^{v_0/2+1} \exp\{-\frac{v_0\ \sigma_{y0}^2}{2\ \sigma_y^2}\}.$$

Here we need to rearrange the likelihood that

$$p(y|d, \sigma_y, s_y) = \sigma_y^{-n} \exp\left\{-\frac{\sum_i y_i^2}{2\sigma_y^2}\right\}$$

$$= (\sigma_y^2)^{-\frac{n}{2}} \exp\{-\frac{n\ v}{2\sigma_y^2}\}$$

where v is the sufficient statistic [2]

$$v = \frac{\sum_i y_i^2}{n}.$$

According to the prior, the likelihood and Eq. (C.2) above, we have (see Fig. 4.1) the full conditional probability of σ_y^2:

$$p(\sigma_y^2|\cdot) \propto p(\sigma_y^2)\ p(y|d, \sigma_y, s_y)$$

$$\propto (\sigma_y^{-2})^{\frac{v_0+n}{2}+1} \exp\{-\frac{v_0\ \sigma_{y0}^2 + n\ v}{2\sigma_y^2}\}.$$

Thus we can conclude that the full conditional probability becomes:

$$(\sigma_y^2|\cdot) \sim Inv - \chi^2\left(v_0 + n, \frac{v_0\ \sigma_{y0}^2 + n\ v}{v_0 + n}\right).$$

[1] if we start with a likelihood distribution with its conjugate prior, the posterior will fall in the same family of distribution as that of the prior.

[2] Define a real value function $T = f(X)$ where X is the observation that generated by the distribution $p(X|\theta)$. We say T is the sufficient statistic if the conditional distribution $p(\theta|T, X) = p(\theta|T)$.

However, the degrees of freedom v_0 in the prior $p(\sigma_y^2)$ is far smaller than the degrees of freedom, n in the likelihood $p(y|d, \sigma_y, s_y)$. So for the sake of of simplicity, the posterior is then approximated by setting $v_0 = 0$, resulting the following non-informative prior [30]:

$$p(\sigma_y^2) \propto \frac{1}{\sigma_y^2}$$

and

$$(\sigma_y^2|\cdot) \propto Inv - \chi^2(n, v). \tag{C.3}$$

C.3 Estimating the Partition Function $z(p_d)$

Here we focus on the estimation of the partition function ($z(s_d)$ in Eq. (4.4), $s_d \in (0..2]$) [44]. One possible approach is to derive from the Bethe free energy. [91] Here we propose an alternative approach based on path sampling [31]. The partition function could be expressed as an integral of the MRF Gibbs density function $q(d|\theta)$ (here $\theta \equiv s_d$) over all the pixel sites:

$$z(\theta) = \int_d q(d|\theta)\mathbf{d}d.$$

To be specific, we are interested in estimating the ratio of the partition function. [31] provides a framework which unifies various approaches toward this goal. Without loss of generality, here we follow [31]'s notation, assume a partition function $z(\theta)$ with θ taking the range of (0..2]. We can represent the Gibbs prior as

$$p(d|\theta) = \frac{q(d|\theta)}{z(\theta)}$$

where d denotes the collection of site variables in the MRF, $q(d|\theta)$

$$q(d|\theta) = \prod_{i\in[1..n]} q(d_i|d_{\partial i}, \theta) \tag{C.4}$$

$$= \frac{1}{\sigma_d^n} \exp\{-\frac{1}{\theta}\sum_{j\sim k} |\frac{d_j - d_k}{\sigma_d}|^\theta\}. \tag{C.5}$$

This is the un-normalized density function with respect to a given θ while $p(d|\theta)$ is the normalized density function. Then taking the logarithm of $z(\theta)$ and differentiating with θ, we can easily obtain the identity (Eq. (6) in [31])

$$\frac{d}{d\theta}\log z(\theta) = \int \frac{1}{z(\theta)}\frac{d}{d\theta}q(d|\theta)\mathbf{d}d = E_\theta[\frac{d}{d\theta}\log q(d|\theta)] \tag{C.6}$$

where E_θ is the expectation with respect to $p(d|\theta)$. To simplify the equations, we derived:

$$\begin{aligned} U(d,\theta) &= \frac{d}{d\theta}\log q(d|\theta) \\ &= \frac{d}{d\theta}\{n\ln\sigma_d - \frac{1}{\theta}\sum_{j\sim k}\rho(\frac{d_j - d_k}{\sigma_d}, \theta)\} \\ &= \sum_{j\sim k, d_j \neq d_k} (\frac{1}{\theta}|\frac{d_j - d_k}{\sigma_d}|^\theta(\frac{1}{\theta} - \ln|\frac{d_j - d_k}{\sigma_d}|). \end{aligned}$$

Notice here we only compute those j, k pixels such that $d_j \neq d_k$, because those $d_j = d_k$ do not make a contribution to this equation, we can simply omit them.

Define the log-ratio as

$$\lambda(a,b) = \log[\frac{z(b)}{z(a)}] \tag{C.7}$$

$$= \int_a^b E_\theta[U(d,\theta)]d\theta \tag{C.8}$$

where $0 < a < b \leq 2$.

As pointed out in [31], Eq. (C.7) can be approximated by numerical integration, e.g., using the trapezoid or Simpson rule. Define j_a and j_b as the indexes such that $\theta_{j_a} \leq a < \theta_{j_a+1} < \ldots < \theta_{j_b-1} < b \leq \theta_{j_b}$. We have, using trapezoid rule, the numerical approximation (Eq. (15) of [31])

$$\begin{aligned}\hat{\lambda}(a,b) = & \frac{1}{2}(\theta_{j_a+1} - a)(\hat{U}_{j_a+1} + \hat{U}_a) \\ & + \frac{1}{2}\sum_{j=j_a+1}^{j_b-2}(\theta_{j+1} - \theta_j)(\hat{U}_{j+1} + \hat{U}_j) \\ & + \frac{1}{2}(b - \theta_{j_b-1})(\hat{U}_b + \hat{U}_{j_b-1})\end{aligned}$$

where $\hat{U}_a$ and $\hat{U}_b$ could be obtained by interpolation. $\hat{U}_i$ is the average of the values of $U(d_i, \theta_i)$ for all simulations draw from θ_i.

Based on these derivations, we come up with the following steps that can be used to estimate the ratio of $z(s_d)$, $s_d \in (0..2]$:

1. For specific s_d^l, s_d^r values, interpolate the existing $\log \frac{z_l}{z_r}$ values to obtain an approximate estimation for $\log \frac{z_l}{z_r}$.

2. Divide the parameter space $z(s_d)$ into even intervals $l = 1, \ldots, L$ and define $z_1 = z(s_d = l)$, $z_L = z(s_d = r)$ with fixed σ_d value.

3. Compute the log-ratio $\lambda(l, r)$ via trapezoid rule.